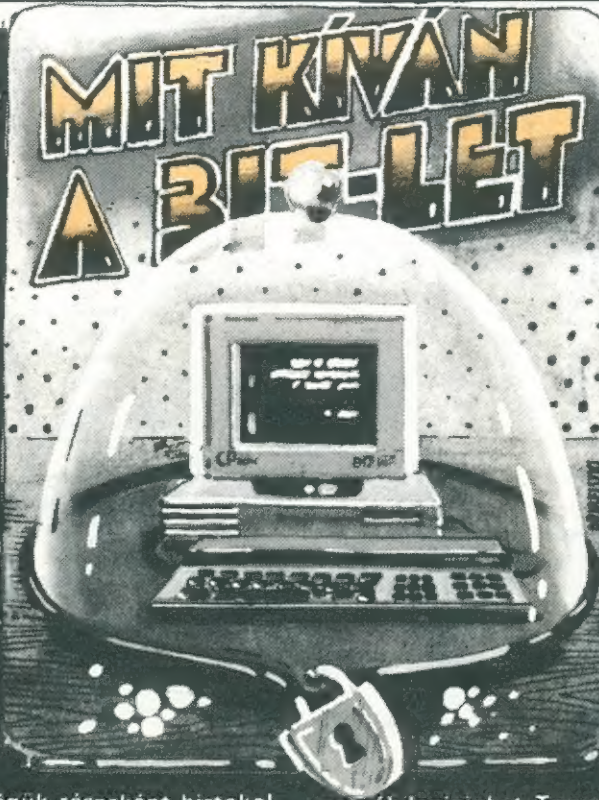


Ismét elkezdődött egy tanév. Talán ezért is kérdezték mostanában többen és többféle hangsúllyal, hogy mit kíván a BIT-LET az új tanévre a diákoknak és a tanároknak, s hogy mit kíván nyújtani a BIT-LET az új tanévben a diákoknak és a tanároknak? Nos, hát a szerkesztő úgy gondolta, hogy megpróbálja néhány sorban összefoglalni régi és új vesszőparipáit, megpróbálja megfogalmazni a számítástechnika oktatásával kapcsolatos kérdőjeleit és felkiáltójeleit.

Elsőként sajnos még mindig azokkal az iskolákkal kellene foglalkoznia az oktatásügy felelőseinek, amelyekben a számítógép mindmáig nem nyerte el megfelelő helyét. Elintézhették persze egy kézlegyintéssel azokat az igazgatókat, akik számára még mindig egy értékes feltárási tárgy a számítógép, amelynek a legjobb helye valamely zárt szekrényben van. Elintézhethetnék ha nem tudnánk, hogy ma már hálistennek ott tartunk, hogy a diák, aki nem találkozik a számítógéppel a felsőoktatás szinte minden területén helyzeti hátrányba kerül azokkal szemben, akiknek természetes dolog a géppel való bánás, akik tehát általános műveltségük részeként birtokolják a tudást, amely a számítógép felhasználásához kell. Jó lenne tehát, ha végre nem intéznék el az ügyet kézlegyintéssel mindazok, akik tehetnek valamit, mindazok, akiknek lehetősége és kötelessége, hogy rászorítsák a fefejűeket, hogy vegyék ki zsebükből a kulcsot, s adják át a gépet a diákoknak.

Második kívánságként egy vágyálmunkat fogalmazzuk meg, miszerint tanulják meg végre a gépek kezelését, használatának lehetőségét azok a pedagógusok is, akiknek semmi közük a matematikához, fizikához, számítástechnikához. S ami úgyszintén elengedhetetlen feltétele a szemléletváltozásnak, hogy legyen minél több gép, legyen legalább 15–20 diákonként egy masina a középiskolákban. Alom mondhatják az olvasók. De ha ez a szám ma még az is, az már tény, statisztikák bizonyítják, hogy azonos anyagi adottságokkal rendelkező iskolák, városok, megyék számítógépellátottsága között hatalmas eltérések vannak, s ezek az eltérések egyszerűen csak arra vezethetők vissza, hogy itt úgyneveztek a dolgot, amott meg inkább másra – fontosabb vagy kevésbé fontosabb ügyekre – költik ugyanazt a pénzt. Aprópos, pénz. Mostanában egyre kevesebben értik, hogy hol is van az a pénz, amelyből a következő években az iskolák gépparkja



gyarapodhat. Amíg központilag osztották szét ezeket az anyagiakat, addig mindenki tudta, hogy hová vesse vigyázó szemét. Ma, amikor talán helyes elgondolásból szétosztották a pénz egy tekintélyes részét, a számítástechnikai oktatók attól tartanak – talán nem is alaptalanul –, hogy a pénz egy része a megyei kasszában irányt változtat.

Kívánjuk még természetesen azt is, amit épp egy éve írtunk le ugyan-ezekben a hasábkokon, hogy a diákoknak ne csak kicsiny része találkozzon a számítógépekkel. De amellett, hogy a legjobbak kiváló programokat írnak, legyen kiforrott módszere és bevált útja annak is, hogy minden diák találkozzon a géppel. Ismerje meg a kezelés mikéntjét, tanuljon róla annyit, amennyi ahhoz kell, hogy ne féljen a gépektől, s értse miként kommunikál az ember és a gép, miként teszi azt a buta, csak egyeket és nullákat értő gép, amit kezelője kíván tőle.

S mit kívánunk magunknak erre az új tanévre? Szeretnénk hasznos segítői lenni továbbra is a szakköröknek, szeretnénk ha tanárok és diákok egyaránt szívesen forgatnák lapjainkat. Tervezzük, hogy megújítjuk és újra állandó rovatunkként kezeljük a Sorvezetőt, amely – talán régebbi pedagógus olvasóink még emlékeznek rá – kifejezetten az iskolai szakkörök munkájához kívánt segítséget adni. Hogy mi lesz benne, hogy mi legyen benne, ebbe jó lenne ha azok is beleszólnának – tudatva velünk elképzeléseiket – akiknek szánjuk a rovatot.

No és természetesen szeretnénk, ha a lap készítésében programjainkkal fölfedezéseikkel továbbra is részt vennének maguk a diákok is. Egyszer már meghirdettük, most újra tesszük, hogy nagyon szívesen közölnénk diákok, szakkörvezető diákok írásait arról, hogy ők hogyan oktatnak, szerintük mi a legcélravezetőbb módszer, tematika a programozás vagy a számítógép-használat tanításában.

Fentiek csak hevenyészett gondolatok, de úgy gondoljuk nem árt, ha időnként egy szerkesztőség nemcsak önmaga számára gondolja végig, hogy mit kíván tenni vagy mit tartana kívánatos állapotnak a lap „környezetét” jelentő kérdésekben. Nem árt ha néha a szerkesztők és az olvasók egymás szemébe is mondják, hogy mit gondolnak egymásról. Ne feledjék, önök is megmondhatják!

Angyalosi László

- 26 **Híroldal** – amelyből azt is megtudhatják, hogy hogyan lehet partnert találni...
- 28 **Programajánlat** – Zene a ZX Spectrumra
- 29 **Notesz** – egy másik Spectrum-program; amely Gépnyerő pályázatunk első fordulójára érkezett
- 32 **Első kézből a tv-computerről** – előzmény a múlt havi számban, folytatás legközelebb
- 34 **Primo – élt 2 évet...** igyekeztünk kideríteni, hogy mi van a pletykák mögött, s hogy lesz-e színes Primo vagy sem?
- 36 **Programajánlat** – Cirill betűk a Primón
- 37 **Hasznos apróságok a C16-ra** – néhány POKE a 64'ER-ből; egy kis adalék a kiselbontású grafikáról írott cikkünkhöz
- 38 **Könyvmoly** – az újonnan megjelent könyvek listájával, egy Spectrum könyv és egy módszeres programozást tanító könyv kritikájával
- 39 **Posta** – amelyben visszatérünk egy régebbi Primo táblázathoz, valamint válaszolunk egy TVC programozó kérdéseire
- 40 **Harmadgépnyerő 3. feladat** – amelyben BIT-LET szigetére látogatunk

Műbőr!

Műbőrt, illetve mesterséges bőrt fejlesztettek ki az egyik robotokkal foglalkozó nyugat-német vállalatnál. A műanyagba ágyazott apró fémgömbök, vagy vezetőképes műanyag szemcsék lehetővé teszik, hogy a robot megfogó szerkezete („keze”) tapintására érzékeljen, és azt követően végezze feladatát. Az új elven működő robotok számos esetben jobban lesznek alkalmazhatók, mint az optikai érzékelés alapján működők.

CHIP csúcs!

A félvezető chipek gyártásában alkalmazott, a szilíciumszeletek fémezésére szolgáló, ún. Pénning-porlasztást tökéletesítette találmányával Kertész Gábor elektromérnök és dr. Vágó György fizikus. Az új eljárással jelentősen megnövekedett a chipek minősége. A Mikroelektronikai Vállalatnál kifejlesztett módszer iránt nagy az érdeklődés és igen jó eredményeket értek el vele az Egyesült Államokban is.

EURÉKA

Mintegy húsz Euréka fejlesztési programba kíván becsatlakozni a jugoszláviai nagy elektronikai cég az Iszakra. A nyugat-európai országok által a csúcstechnológiák elérésére létrehozott félkatonai programhoz azzal a szándékkal csatlakozott az említett jugoszláv vállalat, hogy ily módon biztosítsa versenyképességét a számítástechnikában, az optoelektronikában, a lézerkutatásban és a robotikában.

AMSTRAD

Az angol Amstrad cég PCW 8256 típusjelzéssel egy komplett szövegfeldolgozó, illetve nagy teljesítményű mikroszámítógépet hozott piacra, amely kiegészül egy képernyővel, egy beépített diszkegységgel, billentyűzettel, nyomtatóval és szövegfeldolgozó szoftverrel.

A PCW 8256 teljesen megfelel a szövegfeldolgozási igényeknek. A nagy felbontású képernyőn 90 leütéses, 32 soros szöveg dolgozható fel. A képernyő felülete mintegy negyven százalékban jobban kihasználható, mint más mikroszámítógépeknél. A beépített nagy sebességű RAM diszk az adott dokumentum szerkesztése közben lehetővé teszi az információ gyors tárolását, illetve visszakeresését. A 82 billentyűt tartalmazó billentyűzet speciálisan a szövegszerkesztés igényei szerint készült. A rendszer részét képező nyomtató használható gyors vagy levélminőség üzemmódokban. A PCW mikroszámítógép 256 K memóriával rendelkezik, opcionálisan csatlakoztatható hozzá 1 Mbyte-os lemezegység is.

Kommunikáció!

Svéd szakemberek olyan különleges kommunikációs számítógép rendszert fejlesztettek ki, amellyel egy nyomógombos telefonkészülékkel a világ bármely részéről kapcsolatot lehet teremteni. A rendszer képes venni a beszédhangjeleket és a számítógép által adandó választ beszédre alakítva továbbítja a hívónak. Az új információs, kommunikációs rendszer egyedülálló a világon.

VÁLTOZAT...

A jól ismert Commodore 64 mikroszámítógépet ez év májusától kissé átalakított, modernebb, laposabb formában és új GEOS rendszerrel kiegészítve forgazzák Angliában. A kiegészítéssel az Amiga mikrogéphez hasonlóan lehetőség nyílik menüválasztásra, egérrel vagy botkormánnyal történő működtetésére.

HITACHI

A japán Hitachi óriásvállalatnál már megszokott dolog, hogy minden számítógép kiállításon valami szenzációs OEM-termékkel jelenik meg. Legújabb ilyen produkciója a DK-301 típusú lemezmeghajtója. Ez egy 3 1/2 inches Winchester típusú merevlemez meghajtóegység. Az egész egység mindössze 15 cm mély, 10 cm széles, 3,5 cm magas, de a tárolókapacitása 15 megabájt.

CCOPINFORM

Új számítógép-szerviz nyílt Budapesten a Nefelejcs utca 18-ban. A Commodore gépek és perifériáik, valamint az IBM kompatibilis gépek javításán, karbantartásán kívül vállalják különböző hardverfejlesztések elvégzését, gépek illesztését. S híreink szerint már nyitáskor árusították azt az akusztikus modemet, amely lehetővé teszi számítógépek összekötését a meglévő telefonhálózaton keresztül! Ez utóbbi a BIT-LET Szerkesztőségét is kellően fölízgatta egy akusztikai próba erejéig.

PARTNER

Mikroszámítógépesek információs irodája nyílt a XV. kerületi Csokonai Művelődési Házban. Régi ismerőseink (hiszen velük közösen rendeztük tavaly a BIT-LET Karácsonyt) fejükbe vették, hogy segítenek a géptulajdonosoknak, hogy megtalálják egymást, hogy esetleg megtalálják őket azok, akiknek géptulajdonosokra vagy programozni tudókra lenne szükségük. Aki tehát fölhívja a 690-495-ös telefonszámot vagy személyesen fölkeresi a Csokonai Művelődési Házat a XV., Eötvös u. 64-66-ban, s megadja ott címét, telefonját, azt hogy milyen gépe van, az biztos számíthat rá, hogy a jövőben kap majd különböző információs anyagokat, keresik őt a „márkatársak”, a hasonló érdeklődésű számítógépesek.

FLOPPY KIT

Újabb házilagosan megépíthető hardverkiegészítést kínálnak a hazai amatőröknek ezúttal a Spectrumosoknak. A budapesti Sinclair Klub megállapodást kötött a MICROTEAM GM-mel, hogy a klub tagjai számára megkapja a GM által kifejlesztett floppy illesztő építéséhez szükséges hardverleírást, s a házilag építéshez szükséges egyéb szakmai segítséget. Ez az illesztő bármelyik szabványos floppyval, sőt a Commodore perifériákkal is képes működni, a Spectrum memóriaterületét nem foglalja, s ráadásul lehetővé teszi a Spectrum CP/M-mel való használatát. Az interface építése iránt érdeklődők a Sinclair Klubban érdeklődhetnek a további részletek iránt: minden hétfőn 18-22-ig (csak ilyenkor!) a Szellőző Művek – Budapest XI., Építész u. 14. – klubjában.



U-534

Dán amatőrbúvárok számítógéppel vezérelt robotberendezéssel szeretnék megtalálni a Kattegat öbölben, a második világháború vége felé elsüllyedt U-534 német tengeralattjárót. Az utóbbi időben előkerült náci dokumentumokból kiderült ugyanis, hogy a győzelemben már nem bízó magasrangú német katonai vezetők többek között ezen a tengeralattjárón próbáltak nagymennyiségű aranyat és ékszert kicsempészni. Ebből tervezték fedezni a háború utáni életük anyagi kiadásait.

AMPAR-

Nagy teljesítményű, számítógépes fordítógépet helyeztek üzembe a Szovjetunióban. Az AMPAR névre keresztelt gép óránként 75 oldalnyi gépelt szakszöveget képes angol nyelvről oroszra fordítani. Most dolgoznak a szakemberek a berendezés német, francia és spanyol változatán. A gép elkészítőinek a legnagyobb gondot az okozta, hogy a többjelentésű szavak szövegkörnyezetbe illő jelentését értelmezze és a valódi jelentésnek megfelelően fordítsa le az új fordító berendezés.

utórendelése. Itt van az expedíciós részleg és a számítóközpont is. A mintegy 150 millió készletet a számítógép tartja nyilván és biztosítja, hogy folyamatosan ismert legyen: a 2250 raklap közül melyiken tárolják a keresett könyveket.

Turbo PC

Az amerikai PC's Limited cég bemutatta a Turbo PC elnevezésű új személyi számítógépét. A gép egy 16 bites 8088-2 típusú mikroprocesszorra épül. Központi memóriája 640 kilobyte-os. Tartalmaz egy 360 kilobyte-os lemezegységet, egy 5151 típusú billentyűzetet és egy 135 wattos tápegységet. Az IBM PC-re és PCXT-re írt programok a Turbo PC-n mintegy negyven százalékkal gyorsabban futtathatók. Ára 785 dollár.

SUNNY BOYS!

Az évekkel ezelőtt hihetetlen gyors sikert elért Apple cég, sokféleképpen feldolgozott sikertörténete után, most egy új elektronikai magánvállalkozás került a figyelem középpontjába. A kaliforniai Szilícium-völgyben alig négy évvel ezelőtt alapította a Sun Microsystems céget három alig több mint harmincéves fiatalember. Azóta a napsugár fiúknak (sunny boys-nak) becézett három sikerember az első évi 8 millió dolláros forgalmat 115 millióra, a kezdeti 654 ezer dolláros nyereséget pedig 8,5 millióra növelte. A rohamosan növekvő cég fő profilja az úgynevezett workstation piacra irányul. Olyan nagyteljesítményű professzionális személyi számítógépeket fejlesztenek és gyártanak, amelyek különféle munkahelyeken komplett tudományos, műszaki és grafikus feladatok elvégzésére alkalmasak. A feltörekvő új cég sikerei többek között abban is keresendők, hogy kitűnő minőségű termékeit mintegy húsz százalékkal olcsóbban adja, mint a rivális partnerek.



ÚJ!

Az angliai British Telecom cég új, Qwertyphone elnevezésű készülékében egyesítette a speciális tulajdonságokkal, memóriával rendelkező telefont az adatterminállal és a beépített modemmel. A különleges irodai berendezés főbb jellemzői a következők: 8 K RAM szövegtárolásra, 8 K RAM telefon-számtárlékra, ami 16 K-ra bővíthető. 16 K ROM különféle programok részére. A beépített képernyő egy 4 soros, 32 betűhelyes LCD. A modem 300 Band-os duplex rendszerű. Qwerty billentyűzet programozó és terminálemulációs billentyűvel, továbbá az elkülönített telefonbillentyűvel. A Qwertyphone csatlakoztatható telefonvonalra, soros nyomtatóbemenetre, soros PC billentyűzet-bemenetre stb.

NDK MŰVÉ

A Német Demokratikus Köztársaságban párt- és kormányhatározatok születtek az elektronikai alkatrész és berendezésgyártás nagyarányú fejlesztésére. Két fontos és óriási összegekbe került központi programot hajtának végre az NDK-ban: az egyik a lakásépítés, a másik a mikroelektronika. A ráfordítási összegek nem ismereteseek, de csupán a chipgyártást szolgáló szupertiszta helyiségek létesítése is több milliárd márkába került. Elhatározták, hogy nem vesznek igénybe nyugati alkatrészeket. Elektronikát tartalmazó berendezésekben kizárólag saját, illetve szocialista országokból származó alkatrészeket használnak. Sikeresen gyártják a 8 bites mikroprocesszorokat és dolgoznak a 16 bites elkészítésén. Óriási erőfeszítések árán ma már 8-10%-os kihozatali mutatóval állítják elő a 64 és a 256 K-s RAM-okat.

KÖNYVRAKTÁR

Új számítógépes központi könyvraktár kezdte meg működését a Könyvtérkező és Könyvtárellátó Vállalat Váci úti telepén. A nyolcezer négyzetméter alapterületű raktárban kaptak helyet a könyvjúdonások és sikerkönyvek

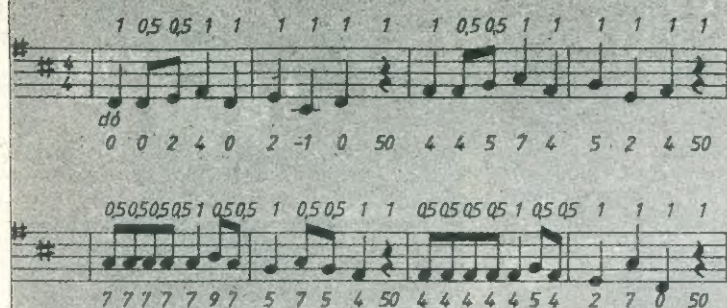
Molekula Bep

A Compodrug Műszaki Fejlesztő Kiszöveget a világon elsőként dolgozott ki a molekulák megjelenítésére alkalmas, szerkezetük tanulmányozását lehetővé tevő, IBM személyi számítógépen futtatható programcsomagot. A dolog jelentősége elsősorban abban van, hogy korábban csak nagyteljesítményű számítógéppel lehetett úgynevezett molekulagrafikai rendszert működtetni.

PROGRAM AJÁNLAT

ZX SPECTRUM
„ZENE”

A kis kakas (5. oszt. 124. old.)



```
5000 REM "Zene"
5005 CLS
5010 PRINT "Új dalt akarsz beírni? - 1": PRINT
5020 PRINT "A régi dalt akarod?": PRINT : PRINT TAB 10; "Lejatszani? - 2"
5025 INPUT v
5030 IF v=1 THEN CLS : RUN 5060
5035 IF v=2 THEN CLS : GO TO 5130
5055 REM "Dal beírása"
5060 INPUT "Hány db hangból áll a dal? ":h: PRINT "Hangok száma: ":h
5065 INPUT "Egységnyi hangérték ideje? ":t: PRINT "Egységnyi hangérték ideje: ":t
5070 INPUT "A dó helye? ":m: PRINT "A dó helye: ":m
5075 DIM T(h)
5080 DIM M(h)
5085 PRINT : PRINT "Add be sorba a hangértékeket!"
5090 FOR i=1 TO h
5095 INPUT T(i): PRINT T(i)
5100 NEXT i
5110 CLS : PRINT "Add be sorba a hangmagasságokat!"
5115 FOR i=1 TO h
5120 INPUT M(i): PRINT M(i)
5125 NEXT i
5130 CLS : PRINT "Nyomj meg egy gombot": PRINT
5135 PRINT "és megszólal a zene."
5140 PAUSE 0
5999 REM "Zene lejátszása"
6000 FOR i=1 TO h
6005 IF M(i)=50 THEN GO TO 6020
6010 BEEP t*T(i),m+M(i)
6015 NEXT i
6020 PAUSE T(i)*50*t
6025 NEXT i
6030 STOP
6100 SAVE "zene"
```

A „ZENE” nevű, ZX Spectrumra írt program alkalmas bármilyen dal, dallam könnyű beírására, újra lejátszására, a „dó” tetszőleges magasságra történő eltolására és a ritmus tetszőleges változtatására.

Mindjárt itt megemlítem, hogy a programot sohasem

indítsuk RUN utasítással, mert akkor a gép töröl minden változót, hanem közvetlen GO TO 5000 utasítással indítsuk!

Ekkor a gép megkérdezi, hogy a korábban beírt dalt akarjuk-e meghallgatni – ekkor a 2-es számot kell betáplálni, majd egy tetszőleges gomb megnyomására megszólal a zene – vagy új dalt szeretnénk beírni: 1-et kell beadni.

Ezek után a gép sorba teszi fel a kérdéseket, melyekhez itt szeretnék néhány magyarázatot adni:

a) A hangok darabszámának beadásánál meg kell számolni, hogy a dal hány db hangból és hány db szünetből áll, és e kettő szám összegét kell beadni. A példaként bemutatott dalnál ez a szám: 41.

b) Az egységnyi hangérték idejénél az ütemmutatónak megfelelően kiválasztjuk, hogy egy egység az negyed, nyolcad stb. lesz, és ennek az egységnyi hangnak mennyi ideig kell szólania (másodpercben). A példánkban: egységnyi hang = negyed = 1, ennek időtartama: 0,5 mp. Ennél a kérdésnél tehát 0,5-et adunk be.

c) A dó helyének megállapításánál szükség van a ZX Spectrum gépkönyvére. Itt találjuk a hangmagasságoknak megfelelő kódszámokat. Példánkban: dó = dé = 2. A beadásra kerülő szám: 2.

A további beírás könnyítése céljából a kotta *főlé* írjuk a hangértékeknek megfelelő számokat (beleértve a szünet értékét is), a kotta *alá* pedig a szolmizálásnak megfelelő hangjegykódokat (úgy, mintha minden dal C-dóban lenne!). A szünetekhez pedig mindig 50-es kódszámot írunk.

A korábbi adatok betáplálása után a gép kéri, hogy először sorban egymás után adjuk be a hangértékeket. Most már csak a kottánk *főlé* írt számokat kell sorba beadni. Ezt követően a gép kéri a hangmagasságokat, itt pedig a kottánk *alá* írt számokat adjuk be.

Ezzel tulajdonképpen be is tápláltuk a gépbe a kívánt dalt. Bármelyik gomb megnyomásával ellenőrizhetjük, hogy mindent helyesen végeztünk-e. Ha igen, akkor az elvárásnak megfelelően szól a dalunk. Ha valami problémát észlelünk (pl. egy hang magassága vagy tartama nem megfelelő), akkor se essünk kétségbe, mert könnyen kijavíthatjuk a hibát. A javításnál megismerjük, hogy hányadik hangot szeretnénk javítani (i), és egy közvetlen utasítással beadjuk a helyes számot:

a hangérték javításakor LET T(i) = ...

a hangmagasság javításakor LET M(i) = ...

Egy közvetlen GO TO 6100 utasítással az egész programot a beírt dallal együtt magnóra vehetjük. Még egyszer hangsúlyozom, hogy visszajátszáskor sohasem RUN-nal, hanem GO TO 5000-rel indítsunk!

A program további lehetősége, hogy az egész dal megváltoztatása nélkül a dó helyét illetve a dal ritmusát tetszőlegesen változtathatjuk. Ezek a változtatások ugyancsak közvetlen utasítással történnek:

A dó helyének áthelyezésekor a dó új helyének megfelelő kódszámot adjuk be (pl. most legyen a dó = gé = 7)

LET m = 7

A ritmus megváltoztatásakor pedig a

LET t = ...

utasításhoz írjuk be az új ritmusnak megfelelő számot. Pl. ha a dalunkat lassítani akarjuk, akkor 0,6-et, ha gyorsítani, akkor 0,4-et írunk.

Ezek az egyszerű változtatási módok nagyon sok lehetőséget kínálnak az ének-zenetanítás területén is.

Remélem, hogy a programom bemutatásával segítséget nyújthattam sok zenét és programozást kedvelő társamnak.

Zátonyi János Szombathely, Váci Mihály u. 9. 9700

NOTESZ

Úgy tűnik, nem mértük föl pontosan, hogy mit vállalunk, amikor kiírtuk a 3 fordulós programozási pályázatot. A C 16 nyerő közel száz programját máig sem tudtuk végignézni. De azt ma már nyugodtan ígérjük, hogy jövő hónapra befejezzük. A noteszokat már mindet láttuk. Vannak köztük egészen jók, de az „igazit” nem találtuk meg. Sokan szem elől tévesztették a legfontosabb szempontot: a felhasználhatóságot. Legnagyobb nehézséget a megfelelő adattárolás és a jó, kényelmes és gyors visszakeresés megoldása okozta.

Általános hibák: sokan a feljegyzés szerinti visszakeresést úgy oldották meg, hogy csak a feljegyzés teljes szövegének pontos beírására működik. Ez szerintünk használhatatlan. Jobb, de nem tökéletes megoldás a kategória szerinti keresés (minden bejegyzést kategorizálni kell, s egy kategóriát lehet egyszerre keresni), ill. a szöveg első akárhány betűje szerinti keresés. A „legprofibbak” noteszában a feljegyzés bármely részlete alapján lehet keresni. Nagyon kevesen csináltak heti összesítőt (rövidített formájú feljegyzésekkel), sokaknál még egy nap programjait se lehet egyszerre a képernyőre íratni – ez is súlyosan rontja a felhasználhatóságot! A feljegyzések kiírásánál sok program nem írja ki a nap nevét, csak a külön naptárfunkcióban azonosítható, hogy május X-e milyen napra esik. Ez kényelmetlen. Legtöbbször az egyéb feljegyzést is szigorú formátumhoz kötötték, pl. csak nevet, címet, telefonszámot lehet feljegyezni. Általános hiba volt még, hogy a feljegyzések beírásakor a gép input rutinját használták, emiatt a feljegyzésbe nem lehet vesszőt stb. írni. Sok, egyébként kitűnő programozó figyelmeztetésre ajánljuk, hogy a legtöbb gépen (pl. C 64) az input rutin használata a felhasználó szemszögéből sokszor kényelmetlen, „profibb” saját input rutint használni (ezt csak egyszer kell jól megírni és számtalan programban lehet használni!) Egy C 64-re írt naptárprogramot a Commodore újság (az Országos Commodore Egyesület lapja) közöl majd. Mi most egy Spectrumra írt naptárprogramot közlünk, mely messze nem tökéletes, de a közölhető és használható Spectrum programok közül a legjobb. Ime tehát *Balassa László* győztesemerei pályázónk programja, a beküldött használati útmutatóval együtt. A program beírásakor egyetlen különlegességre kell figyelni. A program a feljegyzéskategóriák jelölésére nyolc különleges grafikus karakterek használatát. Ezért érdemes a program beírását a 8110–8175 között elhelyezett szubrutin beírásával kezdeni. Ezt a rutint egyszer lefuttatva rendelkezésünkre állnak a speciális karakterek, amelyeket ezután grafikus üzemmódban A–H betűkkel jeleníthetünk meg és írhatunk a lista megfelelő helyére.

ÚTMUTATÓ

a „NOTESZ” program használatához

Általános tájékoztató:

A program ZX Spectrum 48 K gépre íródott. A gépet bekapcsolás után NAGYBETŰS üzemmódba tegyük, és a program teljes futása során ezt a módot használjuk.

A program csak 48 K-s gépen futtatható le.

Amennyiben a program futása leállna, újraindítás GO TO 80-nal vagy GO TO 800-zal.

Általános programismertető:

A „NOTESZ” program tulajdonképpen egy elektronikus határidő-napló, mely egy 100 címes telefonregisztert is tartalmaz. Memóriatakarékossági okokból a programot úgy készítettem el, hogy abba írni, illetve visszakeresni a mindenkori napi dátumhoz viszonyítva 4 hétre előre, illetve egy hétre visszamenőleg lehetséges. A visszakeresés

C-16 NYERŐ
PÁLYÁZAT
PROGRAMJAIT
MEGÍRNI!!

megkönnyítése érdekében beíraskor 8 különféle grafikus szimbólumot lehet használni, melyeket a gép könnyűszerrel felismer, és így a különböző csoportosítások elvégezhetők. Erről bővebben a későbbiekben.

A program használata lépésről lépésre:

A program betöltése után a felhasználó rövid szöveges használati útmutatót olvashat a képernyőn. Innen tovább lépve a program a mai dátum beírását kéri, előbb a hónapot, majd a napot, arab számmal gépelve.

Mit tesz a program a dátum ismeretében?

Kiszámítja, hogy a dátum milyen napra esik, ettől előre 28 napot, visszafelé 7 napot feldátumoz. Megállapítja, hogy a korábbi beírástól a mai napig hány nap telt el, és ennek megfelelően lépteti visszafelé jegyzetünket. Ehhez az átrendezéshez a gépnek pár másodpercre van szüksége, melyhez udvariasan türelmünket kéri. A „feladat” végeztével a képernyőn megjelenik a

MENÜ

- 0 Átlapozás -tól -ig
- 1 egy nap kiírása
- 2 keresés különféle szempontok szerint
- 3 telefon-regiszter
- 4 beírás a jegyzetbe
- 5 SAVE

Nézzük a választéket egyenként:

0 átlapozás -tól -ig (0 gomb)

A „NOTESZ” átlapozását végzi a felhasználó által beírt kezdő dátumtól a végdátumig, naponként, egy teljes jegyzetlap kiírásával. Ha olyan kezdő vagy végdátumot gépelünk be, amely még, vagy már nem szerepel a „NOTESZ”-ban, a program azt nem fogadja el, és helyesbítésre vár.

1 Egy nap kiírása (1-es gomb)

A felhasználó által választott teljes nap jegyzetlapját írja ki a gép a képernyőre.

2 Keresés különféle szempontok szerint (2-es gomb)

Egy tájékoztató jelenik meg a képernyőn, mely tartalmazza a grafikus szimbólumok magyarázatát, ill. azok használatát. Ennek lényege,


```

2 REM "NOTESZ"
3 DIM A$(13,35,27)
4 DIM B$(7,3)
5 DIM C$(12,5)
6 DIM D$(35,59)
7 DIM E$(100)
8 DIM F$(12,5)
9 LET CC=0
10 REM FOCIM
11 PAPER 6: BORDER 6: CLS :
12 PRINT BRIGHT 1;AT 2,0;"Program:"NOTESZ""
13 PRINT AT 5,0; PAPER 7: BRIGHT 1;"A PROGRAM EGY ELEKTRONIKUS HATARIDO NAPLO!"
14 ALFALAD BEIRANDU DATUMTOL SZAMITVA 4 HETRE ELORE IRHATOD BE A NAPI ESEMENYE
15 DET ANAP 24 ORAJARA,TELEFONSZAMOKAT REGISZTERBEN/100 SZAM/TAROLHA- TOD,VALAMI
16 NT MINDEN NAPRA 59 KA-KARAKTERES MEGJEGYZES IS IRHATO. MINDEN ADAT VISSZAKERHETO
17 HES VISSZAMENDELEB 1 HETRE IS"
18 PRINT AT 21,7; FLASH 1; BRIGHT 1;"NOMJ MEG EGY GOMBOT!"
19 PAUSE 0
20 REM ERTEKADAS
21 CLS : PRINT AT 10,10; FLASH 1; PAPER 7: BRIGHT 1;"EGY PILLANAT!"
22 GO SUB 8000
23 GO SUB 8110
24 CLS : PRINT AT 10,7; PAPER 6: BRIGHT 1;"KEREM A MAI DATUMOT!"
25 INPUT PAPER 2; INK 7: BRIGHT 1; FLASH 1;"HONAP/Arab szammal/ ";A
26 INPUT PAPER 1; INK 7: BRIGHT 1; FLASH 1;"NAP/Arab szamal/ ";B
27 CLS : PRINT AT 10,8; PAPER 6: BRIGHT 1;"A GEP A MAI DATUMNAK MEGFELELO- EN"
28 "ATRENDEZI!" A "NOTESZODAT""
29 PRINT AT 12,10; FLASH 1; PAPER 6; INK 7: BRIGHT 1;"KIS TURELMET!"
30 REM A$(FELTOLTESE
31 LET D=0
32 FOR N=1 TO A-1
33 LET D=D+A(N)
34 NEXT N
35 LET C=D+B
36 IF C=D THEN LET D=1: GO TO 350
37 LET D=C-(INT (C/7)+7)
38 IF D=0 THEN LET D=7
39 LET E=A: LET D=D-1: LET P=B-1
40 FOR M=1 TO 26
41 LET D=D+1: LET P=P+1
42 LET A$(1,7+N)=STR$ E+"."+STR$ P+"."+B$(D)
43 IF D=7 THEN LET D=0
44 IF P=A(E) AND E=12 THEN LET E=1: LET P=0
45 IF P=A(E) THEN LET E=E+1: LET P=0
46 NEXT M
47 LET E=A: LET D=D: LET P=B
48 FOR N=7 TO 1 STEP -1
49 IF P=1 AND E=1 THEN LET E=12: LET P=32
50 IF P=1 AND E=7 THEN LET E=E-1: LET P=A(E)+1
51 LET D=D-1: LET P=P-1
52 IF D=0 THEN LET D=7
53 LET A$(1,N)=STR$ E+"."+STR$ P+"."+B$(D)
54 NEXT N
55 REM LEPEKODZ
56 IF C=D THEN CLS : PRINT AT 10,0; PAPER 2; INK 7: BRIGHT 1; FLASH 1;"A REG
57 I DATUM KESOBBI MINT AMIT MAI DATUMNAK BEIRALT!": PAUSE 200: GO TO 100
58 LET L=C-C: LET CC=C
59 REM A$(LEPETESE
60 IF L>34 THEN GO TO 600
61 IF L<0 THEN GO TO 800
62 FOR N=L+1 TO 35
63 LET A$(M,N-L)=A$(M,N)
64 NEXT N
65 LET D=(N-L)=D$(N)
66 NEXT N
67 FOR N=36-L TO 35
68 FOR M=2 TO 13
69 LET A$(M,N)=
70 NEXT M
71 LET D$(N)=
72 NEXT N
73 NEXT N
74 GO TO 800
75 FOR N=1 TO 35
76 FOR M=2 TO 13
77 LET A$(M,N)=
78 NEXT M
79 NEXT N
80 REM MENU KIIRASA
81 PAPER 6: BORDER 6: CLS
82 PRINT AT 3,13; PAPER 7: BRIGHT 1; FLASH 1;"UTMUTATO"
83 PRINT AT 5,3; PAPER 7: BRIGHT 1; FLASH 1;"B":AT 7,3;"1":AT 9,3;"2":AT 11,3;
84 "3":AT 13,3;"4":AT 15,3;"5"
85 RESTORE 819: LET M=1
86 FOR N=0 TO 10 STEP 2
87 LET M=M+1: READ H$
88 PRINT AT N+5,5; PAPER M; INK 9: BRIGHT 1;H$
89 NEXT N
90 DATA "ATLAPOZAS -to?,-ig","EGY NAP KIIRASA","KERESES KLF.SZEMP.SZERINT","TE
91 LEFON REGISZTER","BEIRAS A JEGYZETBE","SAVE"
92 PRINT AT 0,3; FLASH 1; BRIGHT 1;"IRD BE A VALASZTOTT SZAMOT!"
93 FOR N=1 TO 20: PRINT AT 1,N;"#":AT 20,N;"#": NEXT N
94 FOR N=1 TO 20: PRINT AT 1,N;"#":AT N,30;"#": NEXT N
95 PAUSE 0: LET I=INKEY$
96 IF CODE I<48 OR CODE I>54 THEN GO TO 825
97 GO TO 1000+VAL I-5000
1000 REM ATLAPOZAS -TOL,-IG
1005 LET S=0
1010 BRIGHT 0: BORDER 5: PAPER 5: CLS
1015 PRINT AT 7,0; PAPER 2; INK 7: BRIGHT 1; FLASH 1;" HANYADIK HONAP,HANYADIK
1020 NAPJA-TOL KIVAND LATNI A "NOTESZODAT""
1025 INPUT PAPER 7; INK 1; BRIGHT 1; FLASH 1;"HANYADIK HONAP";F
1030 INPUT PAPER 2; INK 7: BRIGHT 1; FLASH 1;"HANYADIKATOL?";FF
1035 CLS : PRINT AT 7,0; PAPER 7: BRIGHT 1; INK 0; FLASH 1;"HANYADIK HONAP,HANYA
1040 DIK NAPJAUIG KIVAND ATLAPOZNI A "NOTESZODAT""
1045 INPUT PAPER 7; INK 0; BRIGHT 1; FLASH 1;"HANYADIK HONAP";G
1050 INPUT PAPER 2; PAPER 7: BRIGHT 1; FLASH 1;"HANYADIK NAPJAUIG?";GG
1055 LET F=STR$ F+"."+STR$ FF+"."
1060 LET G=STR$ G+"."+STR$ GG+"."
1065 FOR K=1 TO 35
1066 IF A$(1,K) ( TO LEN F)=F THEN GO TO 1070
1067 NEXT K
1068 CLS : PRINT AT 10,0; PAPER 0; INK 7: FLASH 1;"AZ ATLAPOZAS KEZDETENEK MEGAD
1069 OTTADATUM NINCSE A "NOTESZBAN"" : PAUSE 350: GO TO 1080
1070 FOR J=1 TO 35
1071 IF A$(1,J) ( TO LEN G)=G THEN GO TO 1070
1072 NEXT J
1073 CLS : PRINT AT 10,0; PAPER 3; INK 7: BRIGHT 1; FLASH 1;"AZ ATLAPOZAS VEGENE
1074 K MEGADOTT DATUM NINCSE A "NOTESZBAN"" : PAUSE 350: GO TO 1030
1075 PAPER 6: BORDER 6: CLS
1076 FOR N=1 TO 3
1077 PRINT AT 1,13; PAPER 0; INK 7: BRIGHT 1; FLASH 1;A$(1,N) ( TO 9)
1078 LET I=1
1079 FOR M=2 TO 13
1080 LET I=I+1: IF I>7 THEN LET I=0
1081 PRINT AT M+2,5; INK 9; PAPER 1; BRIGHT 1;A$(M,N)
1082 NEXT M
1083 PRINT AT 17,5; BRIGHT 1;D$(N)
1084 GO SUB 1300
1085 NEXT M

```

```

1140 CLS : PRINT AT 10,3; PAPER 2; INK 7: BRIGHT 1; FLASH 1;"A "JEGYZETEDET!" A
1145 HOGY KERTED ATLAPOZTUK": PAUSE 300: GO TO 800
1150 REM TABLAZAT RUTIN
1155 PRINT AT 1,7;"Datum": PAPER 0; INK 7: BRIGHT 1;AT 3,0;" Ora:"
1160 FOR V=1 TO 12
1165 PRINT PAPER 7: BRIGHT 1;E$(V)
1170 NEXT V
1175 PRINT AT 17,8; PAPER 0; INK 7: BRIGHT 1;"Megj:"
1180 PRINT AT 17,2; PAPER 7: BRIGHT 1;"#":"A" "#":"B" "#":"C" "#":"D" "#":"E";
1185 AT 20,4;"#":"F" "#":"G" "#":"H"
1190 IF S=1 THEN RETURN
1195 IF S=2 THEN PRINT 0;AT 0,0; FLASH 1; PAPER 2; BRIGHT 1; INK 7;"MASIK ADA
1200 T=M/gomb": GO TO 1317
1205 PRINT 0;AT 0,2; FLASH 1; PAPER 2; INK 7: BRIGHT 1;"LAPOZAS=/L/gomb"
1210 PRINT 0;AT 0,19; FLASH 1; PAPER 2; INK 7: BRIGHT 1;"VESE=/V/gomb"
1215 PAUSE 0: LET I=INKEY$
1220 IF I="L" AND S=0 THEN RETURN
1225 IF I="H" AND S=1 THEN RETURN
1230 IF I="V" THEN GO TO 800
1235 GO TO 1320
1240 REM EGY NAP KIIRASA
1245 GO SUB 1503
1250 PAPER 5: BORDER 5: CLS
1255 LET S=1
1260 PRINT AT 10,1; PAPER 6; INK 0; BRIGHT 1;"MELYIK NAPOT KIVAND LATNI?"
1265 INPUT FLASH 1; PAPER 7: BRIGHT 1;"HANYADIK HONAP? ";F
1270 INPUT FLASH 1; PAPER 7: BRIGHT 1;"HANYADIKAT? ";FF
1275 LET I=F+STR$ F+"."+STR$ FF+"."
1280 FOR N=1 TO 35
1285 IF A$(1,N) ( TO LEN F)=F THEN GO TO 1550
1290 NEXT N
1295 CLS : PRINT AT 10,1; PAPER 2; INK 7: BRIGHT 1; FLASH 1;"A KERT NAP NINCSE A
1300 "NOTESZBAN""
1305 PAUSE 300: RETURN
1310 PAPER 6: BORDER 6: CLS
1315 PRINT AT 1,13; PAPER 0; INK 7: BRIGHT 1;A$(1,N) ( TO 9)
1320 LET I=1
1325 FOR M=2 TO 13
1330 LET I=I+1: IF I>7 THEN LET I=0
1335 PRINT PAPER 1; INK 9; BRIGHT 1;AT M+2,5;A$(M,N)
1340 NEXT M
1345 PRINT AT 17,5; BRIGHT 1;D$(N)
1350 GO SUB 1300
1355 IF S=2 THEN RETURN
1360 GO TO 1500
1365 STOP
1370 REM KERESSES
1375 BORDER 5: PAPER 5: CLS
1380 GO SUB 8300
1385 CLS
1390 PRINT PAPER 6; BRIGHT 1;AT 7,0;"IRJ BE EGY SZIMBOLUMOT ES A GEP KIIRJA JEG
1395 ZETED AZT A SORAT MELYET EZZEL A SZIMB.-AL IRTAL "
1400 BRIGHT 1: GO SUB 8522: BRIGHT 0
1405 INPUT PAPER 1; BRIGHT 1; INK 7: FLASH 1;"KEREM A SZIMBOLUMOT ";J$
1410 IF CODE J$<144 OR CODE J$>151 THEN GO TO 2020
1415 PAPER 6: CLS
1420 PRINT 0;AT 1,7; PAPER 2; INK 7: BRIGHT 1; FLASH 1;"KIS TURELMET KEREKI!"
1425 LET Y=0
1430 FOR M=2 TO 13
1435 FOR K=1 TO 5
1440 POKE 23692,255
1445 IF A$(M,N) (K)=J$ THEN PRINT TAB 0; PAPER 0; INK 7: BRIGHT 1;A$(1,N) ( TO 9)
1450 ; PAPER 7: BRIGHT 1; INK 0;E$(M-1); PAPER 6;A$(M,N)
1455 POKE 23692,255
1460 IF D$(N) (K)=J$ AND Y=0 THEN PRINT TAB 0; PAPER 0; INK 7: BRIGHT 1;A$(1,N) (
1465 TO 9); ; PAPER 7: BRIGHT 1; INK 0;"Megj:"; PAPER 6;D$(N); LET Y=1
1470 NEXT K
1475 NEXT M
1480 NEXT N: INPUT ""
1485 PRINT 0;AT 0,0; PAPER 1; INK 7: BRIGHT 1;"A "NOTESZBAN"" NINCSE TOBB PROGR
1490 AM":AT 1,10; PAPER 7; INK 0; FLASH 1;J$; FLASH 0; PAPER 1; INK 7;"-AL JELELVE"
1495 PAUSE 200
1500 INPUT ""
1505 LET S=3: GO SUB 1312
1510 CLS : GO TO 2010
1515 STOP
1520 GO TO 7000
1525 REM BEIRAS A JEGYZETBE
1530 LET S=2
1535 BORDER 5: CLS
1540 GO SUB 8500
1545 BORDER 5: PAPER 5: CLS : PRINT AT 10,0; PAPER 2; INK 7: BRIGHT 1;"NOTESZ
1550 O" MELYIK "LAPJARA" KI- ";AT 11,5;"VANSZ ADATOT BEIRNI?"
1555 GO SUB 1510
1560 IF D=3 THEN GO TO 3015
1565 INPUT PAPER 7: BRIGHT 1;AT 0,0; FLASH 1;"MELYIK ORA?/CSAK AZ ELBO SZAMOT I
1570 RD BE!/MEGJEGYZESBE IRNI=30 "IT
1575 PRINT PAPER 7: BRIGHT 1;AT 0,0; FLASH 1;"IRD BE A JEGYZETEDET,A SZIMBOLU-M
1580 OKKAL VAGY ANELKUL! ";H$
1585 LET D=0
1590 FOR K=0 TO 22 STEP 2
1595 LET O=O+1
1600 IF K=T THEN LET A$(O+1,N)=H$: GO TO 3065
1605 NEXT K
1610 IF T=30 THEN LET D$(N)=H$: GO TO 3065
1615 GO SUB 1550
1620 GO SUB 1312
1625 GO TO 3015
1630 STOP
1635 GO TO 9990
1640 STOP
1645 REM TELEFON REG.
1650 PAPER 3: BORDER 3: CLS
1655 PRINT AT 4,12; FLASH 1; PAPER 6; BRIGHT 1;"VALASSZ!"
1660 PRINT AT 7,3; PAPER 7: BRIGHT 1; INK 0; FLASH 1;"U":AT 7,5; FLASH 0; PAPER
1665 6;"UJ TELEFONSZAM BEIRASA":AT 9,3; PAPER 7: FLASH 1;"K":AT 9,5; FLASH 0; PAPER 6
1670 ;"TELEFONSZAM KERESSESE":AT 11,3; PAPER 7: FLASH 1;"T":AT 11,5; FLASH 0; PAPER 6;
1675 ;"TELEFONREGISZTER KIIRASA"
1680 PRINT AT 10,6; PAPER 0; INK 7: BRIGHT 1;"U/vagy/K/vagy/T/gomb"
1685 PAUSE 0: LET I=INKEY$
1690 IF I="U" THEN GO TO 7300
1695 IF I="K" THEN GO TO 7050
1700 IF I="T" THEN GO TO 7200
1705 GO TO 7020
1710 REM szam keresese
1715 BORDER 3: PAPER 3: CLS : PRINT AT 10,1; FLASH 1; PAPER 2; INK 7: BRIGHT 1;"
1720 IRD BE KIVAND SZAMAT KERESSED?"
1725 INPUT PAPER 7: BRIGHT 1; INK 0;"VIGYAZZ! PONTOS NEVET IRJ! ";J$
1730 IF LEN J$>15 THEN LET J$=J$ ( TO 15)
1735 FOR N=1 TO 100
1740 IF C$(N) ( TO LEN J$)=J$ THEN CLS : PRINT AT 6,0; PAPER 1; INK 7: BRIGHT 1;
1745 C$(N);AT 8,10; PAPER 2; BRIGHT 1; INK 7;"HIVOSZAMAS":AT 10,12; PAPER 0; INK 7; F
1750 LASH 1;B(N); GO TO 7085
1755 NEXT N
1760 CLS : PRINT AT 10,0; PAPER 0; INK 7: BRIGHT 1; FLASH 1;"ILYEN NEV NINCSE A R
1765 EGISZTERBEN!": PAUSE 150: GO TO 7055
1770 PRINT 0;AT 0,7; PAPER 7; INK 0; BRIGHT 1;"MASIK SZAM=/M/gomb"
1775 PRINT 0;AT 1,7; PAPER 7; INK 0; BRIGHT 1;"VISSZA=/V/gomb":AT 2,7;"MENU=/SP

```


NOTESZ

```

ACE/gomb"
7070 PAUSE 0: LET I$=INKEY$
7095 IF I$="M" THEN GO TO 7055
7100 IF I$="V" THEN GO TO 7085
7102 IF I$="CHR$ 32 THEN GO TO 800
7105 GO TO 7090
7200 REM KIÍRAS
7202 BORDER 5: PAPER 5: CLS
7203 GO SUB 7325
7210 GO TO 800
7299 STOP
7300 REM BEÍRAS
7302 CLS : FOR N=1 TO 100
7305 IF C$(N) ( TO 1)="" THEN LET R=N: LET S=1: GO TO 7380
7310 NEXT N
7315 LET S=0: PRINT AT 10,0: PAPER 1: INK 7: BRIGHT 1:"NINCS A REGISZTERBEN SZABAD HELY":AT 12,3: PAPER 0: FLASH 1:"NEID AT A REG.-T ES IRJ EGY ":AT 13,4:"REGIADAT HELYERE UJAT:"
7320 PAUSE 250: CLS : GO SUB 7325
7322 GO TO 7375
7325 FOR N=1 TO 100 STEP 20
7326 PRINT AT 0,6: PAPER 2: INK 7: BRIGHT 1:"Előfizeto neve":AT 0,23:"T.szám"
7327 FOR M=0 TO 19
7330 PRINT PAPER 7: BRIGHT 1: INK 0:AT M+2,N+M: ".":TAB 6:C$(N+M):" " "AT M+2,N+M)
7337 NEXT M
7339 GO TO 7350
7340 NEXT N
7345 GO TO 7368
7350 PRINT #1:AT 0,7: PAPER 0: INK 7: BRIGHT 1:"TOVABB=T/gomb":AT 1,7:"STOP=S/gomb"
7355 PAUSE 0: LET I$=INKEY$
7360 IF I$="T" THEN CLS : GO TO 7340
7365 IF I$="S" THEN RETURN
7367 GO TO 7355
7368 PRINT #1:AT 0,4: PAPER 7: BRIGHT 1: INK 0:"A LISTA VEGERE ERTUNK!":AT 1,10: PAPER 1: INK 7: BRIGHT 1:"UJRA=U/gomb"
7369 PAUSE 0: IF INKEY$="U" THEN INPUT " ": GO TO 7325
7372 GO TO 7369
7375 INPUT BRIGHT 1: INK 0: PAPER 6:"IRD BE MELYIK SORSZAMRA KERUL AZ UJ SZAM? ":R
7380 CLS : INPUT PAPER 1: BRIGHT 1: INK 7: FLASH 1:"ELOFIZETO NEVE: ":C$(R)
7385 INPUT PAPER 2: BRIGHT 1: INK 7: FLASH 1:"TELEFONSZAM? ":B(R)
7387 CLS : GO SUB 7500
7390 CLS : PRINT AT 0,4: PAPER 7: BRIGHT 1: FLASH 1:"U": FLASH 0: PAPER 6:"UJ A DAT BEÍRÁS"
7395 PRINT AT 10,4: PAPER 7: BRIGHT 1: FLASH 1:"V": FLASH 0: PAPER 6:"VISSZA A T.REGISZTERHEZ"
7400 PRINT AT 12,4: PAPER 7: BRIGHT 1: FLASH 1:"F": FLASH 0: PAPER 6:"FOPELOSZTASRA /menu/"
7405 PRINT AT 4,2: PAPER 2: INK 7: FLASH 1: BRIGHT 1:"IRD BE A MEGFELELO KARAKTER"
7410 PAUSE 0: LET I$=INKEY$
7420 IF I$="U" THEN GO TO 7302
7425 IF I$="V" THEN GO TO 7085
7430 IF I$="F" THEN BRIGHT 0: GO TO 800
7435 GO TO 7410
7500 REM BETURENDBE RENDEZES
7505 PRINT AT 10,0: PAPER 6: BRIGHT 1: INK 0:"A TELEFONKÖNYVET BETURENDBE RENDEZEL A GEP,EGY KIS TURELMEZ KER!"
7510 FOR N=1 TO 100
7520 IF C$(N) (1)="" THEN GO TO 7540
7530 NEXT N
7535 LET N=100
7540 FOR M=1 TO N-1
7545 LET Z=0
7550 FOR K=1 TO N-M
7555 IF C$(K)=C$(K+1) THEN GO TO 7580
7560 LET K=C$(K): LET X=B(K)
7565 LET C$(K)=C$(K+1): LET B(K)=B(K+1)
7570 LET C$(K+1)=K$: LET B(K+1)=X
7575 LET Z=1
7580 NEXT K
7585 IF Z=0 THEN GO TO 7600
7590 NEXT M
7600 CLS : RETURN
7699 STOP
8000 REM B$( ) A( ) E$( ) FELTÖLTÉSE
8005 RESTORE 8050
8010 FOR N=1 TO 7: READ B$(N): NEXT N
8015 RESTORE 8060
8020 FOR N=1 TO 12: READ A(N): NEXT N
8025 RESTORE 8070
8030 FOR N=1 TO 12: READ E$(N): NEXT N
8035 DATA "SE", "CSU", "PEN", "SZO", "VAS", "KET", "KED"
8040 DATA 31,28,31,30,31,30,31,31,30,31,30,31
8070 DATA "00-02", "02-04", "04-06", "06-08", "08-10", "10-12", "12-14", "14-16", "16-18", "18-20", "20-22", "22-24"
8100 RETURN
8110 REM UDG
8115 RESTORE 8140
8120 FOR N=0 TO 63
8125 READ D
8130 POKE USR "A"+N,D
8135 NEXT N: RETURN
8140 DATA 0,252,32,46,40,44,40,46
8145 DATA 0,240,34,36,42,42,46,42
8150 DATA 0,48,72,64,55,10,74,55
8155 DATA 54,73,73,65,34,20,0,0
8160 DATA 144,208,176,144,153,13,11,9
8165 DATA 130,146,162,194,162,146,138,0
8170 DATA 70,169,136,134,129,169,70,0
8175 DATA 98,146,138,98,18,146,98,0
8500 REM TAJEKOZTATÓ RUTIN
8505 PAPER 7: BORDER 6: BRIGHT 1: CLS
8510 PRINT AT 0,4: PAPER 2: INK 7: BRIGHT 1:"TAJÉKOZTATÓ A BEÍRÁSHOZ":AT 1,5:"ES A VISSZAKERESSEHEZ"
8520 PRINT AT 2,0:"Ha használatod napi feladataid rögzítésénél és visszakeresésénél az alábbi szimbólumokat akkor a gép azokat felismerve kitűdja keresni az ilye n irányú elfoglaltságaidat.Ezen kívül természetesen bármint írhatasz a megfelelő nap,megfelelő órájához max. 27 kar. hosszban,ill.a megjegyzés rovatba 59 ka r.hosszban."
8521 PRINT AT 12,0: FLASH 1: INVERSE 1:"A SZIMBOLUMOKAT MINDIG A 'SOR EL-SO 5 HELYENKÉ VÁLAMELYIKERE TEDD"
8522 RESTORE 8535
8524 FOR N=0 TO 7: READ H$
8525 PRINT "GRAF. ":CHR$(65+N):" ": PAPER 6:CHR$(144+N): PAPER 7: " ": PAPER 2: INK 7:H$
8530 NEXT N
8535 DATA "FONTOS TELEFON", "TÁRGYALAS,MEGBESZÉLES", "SZÜLETESNAP", "TALÁLKOZD A KE DVESSSEL", "HELVNAP", "KIKÜLDÉTES", "CSALADI PROGRAM", "SZABADIDŐ,SZORAKOZÁS"
8540 PRINT #1: FLASH 1: PAPER 0: INK 7: BRIGHT 1:AT 0,5:"NYOMJ MEG EGY GOMBOT!"
8545 PAUSE 0: BRIGHT 0: RETURN
9989 REM SAVE-VERIFY
9990 CLS : PRINT AT 10,13: PAPER 2: INK 7: BRIGHT 1: FLASH 1:"SAVE": SAVE "NOTES Z"
9994 CLS : PRINT AT 10,7: PAPER 6: BRIGHT 1: FLASH 1:"NYOMJ MEG EGY GOMBOT!"
9995 PAUSE 0: CLS : PRINT AT 10,12: PAPER 1: BRIGHT 1: INK 7: FLASH 1:"VERIFY": VERIFY "NOTESZ"
9997 GO TO 800
9999 REM BALASSA LASZLO GYORSZE-MERE MELYFUROTELEP 1986 ZX SPECTRUM-48K

```



hogy különféle grafikus szimbólummal jelölhetjük meg feljegyzéseinket. Ezek:

- fontos telefon
- születésnap
- névnap
- randevő
- fontos tárgyalás
- kiküldetés
- szabadidő, sport
- családi események, programok

Megfelelő grafikus szimbólum begépelése után a gép kigyűjti azokat az elfoglaltságainkat, melyet ezen jellel különböztettünk meg beírásunknál.

3 Telefonregiszter (3-as gomb)

A telefonregiszterünk 100 hívószámot és nevet képes tárolni.

Telefonregiszter használata:

MENÜ:

- Új szám beírása
- Már beírt szám keresése
- A teljes regiszter kiírása

a) Új szám beírása:

A program megállapítja, van-e még szabad hely a regiszterben; ha nincs, kiírja a regiszter tartalmát, és kéri, hogy egy régi hívószám törlésével biztosítsa a felhasználó helyet az új, beírandó adatnak.

b) Keresés:

A program lehetőséget nyújt, hogy az előfizető nevének beírásával a géppel keressük ki a regiszterből a megfelelő hívószámot.

c) Teljes regiszter kiírása:

A teljes telefonkönyvünk kiírására nyílik mód a program ezen részén.

4 Beírás a jegyzetbe (4-es gomb)

A képernyőn ismét megjelenik a szimbólumhasználati tájékoztató, melynek elolvasása után meg kell jelölni, hogy milyen dátumra kívánunk beírni eseményt.

Ezt követően a gép kiírja a kérdéses napot, melyből kitűnik, hogy ezen ideig milyen elfoglaltságot jegyeztünk be erre a napra. A megfelelő óra beírását vagy a 30-as szám begépelését (mely a megjegyzés rovatba írja a szöveget) ezt követően a szimbólumot és a szöveget kell beírni. Az ENTER sorzáró lenyomása után ismét megjelenik a kérdéses nap, de most már a legfrissebb beírásunkat tartalmazza.

Figyelem!

Az órák párosával vannak feltüntetve, mindig az elől álló órát kell megjelölni. Egy órához 27 karakternyi szöveg írható. A megjegyzésbe 59 karakter hosszúságú szöveg írható.

5 SAVE (5-ös gomb)

Programunk jegyzetünk kimentését végzi. A SAVE befejeztével a magnót lejátászásra állítva egy gomb megnyomása után a gép VERIFY funkciót végez.

Hibátlan másolás esetén a képernyőn a **MENÜ** jelenik meg.



Mostanában egyre több, a TV Computerrel kapcsolatos kérdést kapunk. Némi fej-törés után – vajon ki vála-szolhatna ezekre? Úgy dön-töttünk, hogy megkeressük a legilletékesebbeket. Így hát a gép fejlesztésében részt vett fiatal mérnökökkel „kötöttünk egyezséget”, mi-szerint ellátják olvasóinkat információkkal. A múlt havi számban közöltük Cseh Ti-bor írásának első felét, íme a második rész.

ELSŐ KÉZBŐL

A TV COMPUTER RŐL

LOGIKAI ESZKÖZ, FUNKCIÓOSZTÁLY, HOZZÁRENDELÉSI TÁBLÁZATOK

Az operációs rendszer rutinjai egy-egy logikai eszköz ki-szolgálását, kezelését végzik. A logikai eszközök rendelteté-süktől függően különböző számú szubrutinnal rendelkez-nek, melyek közül a legfontosabb a karakteres és blokkos input és output.

A logikai eszközök számozása 0-tól 7-ig terjed az alábbiak szerint:

- 0 – **video**: alapvetően a képernyőre rajzolás a feladata. Emellett tartalmazza a képernyőtörlő, a grafikus fel-bontást és palettaszíneket beállító és a karakterdefini-áló rutinokat.
- 1 – **billentyűzet**: a billentyűzet leolvasását végzi.
- 2 – **editor**: biztosítja az egész képernyős programszer-kesztés (full screen editor) lehetőségeit, ellátja a konzol input-output funkciókat. Az input felhasználja (meg-hívja) a billentyűzet karakterolvasó rutinját.
- 3 – **hang**: a hangkeltő áramkört vezérli.
- 4 – **nyomtató**: a párhuzamos nyomtató kiszolgálását végzi.
- 5 – **kazetta**: a magnókazettán történő program- és adat-tárolást, illetve annak visszaolvasását hajtja végre. Tartalmazza az ellenőrző (VERIFY) rutint is.
- 6 – **bővítőkártyák**: az operációs rendszer csak az RS232 soros vonali kártya kezelőprogramját tartalmazza. Ha van ilyen kártya bedugva a TVC-be, akkor bekapcso-láskor ez lesz a kijelölt logikai eszköz.
- 7 – **Kernel**: a logikai eszközök és a funkcióosztályok egy-máshoz rendelését végzi.

A logikai eszközökhöz csak közvetett módon, hozzárende-lési táblázatokon keresztül lehet hozzáférni. Két hozzáren-delési táblázat van, egy az input, egy pedig az output rend-szerhívásokhoz. Mindkét táblázat 8 elemű, a rekeszek szá-mozása 0-tól 7-ig történik. A táblázatok egyes rekeszei az input és output funkcióosztályokat reprezentálják, a rekesz-ben pedig annak a logikai eszköznek a száma van, amelyik az adott funkcióosztály rutin hívásait kiszolgálja.

A funkcióosztályokat is elnevezhetjük sorszámuk szerint a logikai eszközök neveivel: 0–video, 1–billentyűzet stb. Alapvetően igaz az, hogy a funkcióosztályok megfelelnek az egyes logikai eszközöknek, tehát a 0 funkcióosztály tartozik a video hívásokhoz, az 1. osztály a billentyűzet-hez stb.

A hozzárendelési táblázatok vége „nem szabályos”: a 7. osztály mindig a Kernel hívásokat fogja jelenteni, emiatt az input és output táblázatban a 7. osztálynak nincs rekesze! A táblázatok utolsó eleme a bővítőkártyák osztályához tar-tozik és azt mutatja, hogy a négy közül (0–3) melyik aljzat-ban van az aktuálisan kijelölt bővítőkártya. Az aljzatok közül a jobboldali, azaz a legfelül levő a 0. számú.

Az input és output hozzárendelési táblázat kezdő címe:

IN TABLE: 2816=0B00h
OUT TABLE: 2824=0B08h

Bekapcsoláskor a következő értékek kerülnek a táblázatokba:

| | | |
|------------|------|------------------------------|
| IN TABLE: | DEFB | OFFh ; video |
| | DEFB | 1 ; billentyűzet |
| | DEFB | 2 ; editor |
| | DEFB | OFFh ; hang |
| | DEFB | OFFh ; nyomtató |
| | DEFB | 5 ; kazetta |
| | DEFB | 6 ; bővítőkártyák |
| | DEFB | OFFh ; aktuális bővítőkártya |
| OUT TABLE: | DEFB | 0 ; video |
| | DEFB | OFFh ; billentyűzet |
| | DEFB | 2 ; editor |
| | DEFB | OFFh ; hang |
| | DEFB | 4 ; nyomtató |
| | DEFB | 5 ; kazetta |
| | DEFB | 6 ; bővítőkártyák |
| | DEFB | OFFh ; aktuális bővítőkártya |

Ha egy rutin hibajelzéssel tér vissza, akkor a hívott funkció-osztályt visszarendeli a saját logikai eszközehez az operációs rendszer.

A logikai eszközök legfontosabb feladata a karakteres és blokkos input illetve output. Amelyik osztály saját eszköze nem tudja ezt végrehajtani, ott a táblázatban OFFh érték áll, egyéb hívásokra (ha van ilyen) a OFFh érték a saját eszközt jelöli ki. Az utolsó elem helyén a OFFh definiálatlan bővítőkártya kijelölést mutat.

A funkcióosztályok és logikai eszközök hozzárendelése megváltoztatható. Egyik lehetőség erre a Kernel rutinok hívása, a másik pedig a hozzárendelési táblázatok közvetlen átírása. A hozzárendelési táblázatoktól függetlenül a Kernel osztályhoz a Kernel rutinok tartoznak, sőt a Kernel rutinokat tilos más osztályokhoz hozzárendelni!

A hozzárendelések megvalósításának három típusa van:

- 1. I/O átrendelés**: a karakteres és blokkátvitelt rendeli egy másik eszközhöz. A megfelelő eszköz számát (0–5) kell a kívánt táblaelembe írni. Az osztály többi hívását a saját eszközeinek rutinjai szolgálják ki.
- 2. I/O átrendelés egy bővítőkártyához**: a karakteres és blokk átvitelt egy bővítőkártyához rendeli. A bővítő-kártya eszköz számát (6) kell a kívánt táblaelembe írni, valamint a táblázat utolsó elemébe annak az aljzatnak a számát (0–3), amelyikbe a kártya be van dugva. Az osztály többi hívását a saját eszközeinek rutinjai szol-gálják ki.
- 3. Közvetlen hozzárendelés egy bővítőkártyához**: a funkcióosztály összes hívását egy bővítőkártyához rendeli.


```

10 DEF HEX$(X)=CHR$(X+48-7*(X>9))
12 DEF HEX2$(X)=HEX$(INT(X/16)) & HEX$(X AND 15)
14 DEF HEX4$(X)=HEX2$(INT(X/256)) & HEX2$(X AND 255)
16 DEF DEC(X$)=ORD(X$)-48+7*(ORD(X$)>57)+32*(ORD(X$)>96)
18 DEF DEC2(X$)=16*DEC(X$(1)) + DEC(X$(2))
20 DEF DEC4(X$)=256*DEC2(X$(1:2)) + DEC2(X$(3:4))

100 !-----!
102 ! !
104 ! TV-COMPUTER 2.sz. program !
105 ! !
106 ! Decimális-hexa átszámítás !
107 ! !
108 !-----!
110 GRAPHICS 4
120 PRINT " Decimális számok hexa értéke"
130 PRINT
140 I=0
150 FOR J=1 TO 4
160 : GOSUB 200
170 NEXT J
180 END
190 :
200 !-----!
202 ! !
204 ! 64 bájt átváltása !
206 ! !
208 !-----!
210 FOR I=1 TO I+63
220 : I$=HEX2$(I)
230 : PRINT USING "####":I;
240 : PRINT USING "=##h":I$;
250 NEXT I
260 PRINT
270 GET
280 RETURN

```

Az alapgondolat: az editor input osztályt hozzárendeljük a kazettához, és a billentyűzet helyett a listát tartalmazó adatfile-ből beolvassuk a programsorokat. A BASIC interpreter a beolvasott sorokat beilleszti a memóriában tárolt programba.

Nézzük meg pl. a 2. sz. programot. A 10–20 sorokban van hat függvénydefiníció, melyek a decimális-hexa konverziót hajtják végre. Csináljunk ebből a MERGE számára listát egy kazettás adatfájlba:

OPEN OUTPUT "HEXDEC"

ok

A magnót a következő sor lezárása előtt kell elindítani:

LLIST#5:10–20:CLOSE OUTPUT

ok

Az LLIST hatására előbb a fejléc, azután 256 bájtos blokkban a lista kerül a szalagra. Az utolsó blokkot csak a CLOSE hatására küldi ki a BASIC! Ezzel kész a kazettás lista.

Most jöhet a MERGE. Töltsünk be egy másik programot, vagy egyszerűen csak írjuk át a 10–20 sorokat pl. üres PRINT-ekre. Megnyitjuk a listafájl a kazettán és hozzárendeljük az editor inputot a kazettához:

OPEN "HEXDEC":POKE 2818,5

Searching

Reading: HEXDEC

ok

ok

Az OPEN hatására megjelenik a Searching felirat. Amikor a fejléc blokkot beolvasta, kiírja: Reading: HEXDEC. Az OPEN hatására csak az első, azaz a fejléc blokk beolvasása történik meg. A POKE átírja a hozzárendelési táblában az editor input osztályt a kazettához. A POKE végrehajtását jelzi az első "ok" a képernyőn és a kurzor még nem látható. Ezután megtörténik az adatblokkok beolvasása. A beolvasott sorokat ugyanúgy ahogy a billentyűzetről beírt sorokat, beilleszti a BASIC a memóriában levő programba.

Ha vége van az adatfile-nak, akkor a "file vége" hibajelzés hatására az operációs rendszer visszarendeli az editor input osztályt az editorhoz, kiírja a második "ok" üzenetet és újra látható lesz a kurzor. Már csak egy dolog van hátra:

CLOSE

ok

Ezzel a MERGE kész, erről listázással meggyőződhetünk.

Figyelmeztetés: Aki nem távvezérléses magnóval dolgozik, annak a hosszabb programok összefésülése első nekifutásra hibajelzéssel le fog állni. Ennek oka, hogy a beolvasott programsorokat a BASIC interpreternek értelmeznie és tömörítenie kell, ami esetenként több időt vesz igénybe, mint amennyi két adatblokk között rendelkezésre áll. (Ilyenkor tanácsos a magnó „pillanat áll” gombjával néhány másodperces szüneteket beiktatni az adatblokkok beolvasása közé.)

A kívánt táblaelembe a legfelső bit 1-be állításával annak az aljzatnak a számát kell beírni (80h–83h), amelyikbe a kártya be van dugva.

A 3. típusú hozzárendelésre példa a floppy csatoló kártya használata. Ha ez be van dugva a TVC-be, akkor bekapcsoláskor az input és output kazettaosztály közvetlenül a floppyhoz lesz rendelve, azaz minden kazettafunkciót a floppykezelő rutinok hajtanak végre.

A BASIC leírásban szereplő „periféria” megfelel az itt ismertetett funkcióosztálynak. A PRINT utasítás a 2. számú kijelölt perifériára dolgozik, ez az editor funkcióosztályt jelenti. Az LPRINT utasítás kijelölt perifériája a 4. számú, ez a nyomtató osztály. Bizonyos BASIC utasításokban meg lehet adni, hogy a végrehajtás melyik periféria meghívásával, azaz melyik funkcióosztállyal történjen: pl. a PRINT"TVC" a képernyőre ír, míg a PRINT#4:"TVC" a nyomtatóra. Hasonlóan az LPRINT"TVC" a nyomtatóra ír, az LPRINT#2:"TVC" pedig a képernyőre. Ez a paraméterezés nem változtatja meg a hozzárendelési táblázatok elemeit, hanem másik funkcióosztály meghívásával fog működni.

PÉLDA A HOZZÁRENDELÉSI TÁBLA ÁTÍRÁSÁRA: MERGE ELJÁRÁS

A TVC 1.2 BASIC verzióban nincs MERGE utasítás, ennek ellenére a MERGE funkció, azaz programok összefésülése megoldható a hozzárendelési tábla átírásának segítségével.



A PRIMO-nak az első perctől kezdve szurkoltunk. Hiszen ez a magyar mikrogép akkor szállt be a ringbe, amikor már ott voltak a nagy hirű és múltú ellenfelek: a Commodore, a Spectrum, a HT-vé lett TRS. Örültünk amikor egy igazán jó Primoval találkoztunk, a színes, nyomógombos Pro-Primoval. Akkor, néhány hónappal ezelőtt azt hittük, hogy most kezdődik a gép igazi karrierje. Sajnos tévedtünk.

„A Primo-nak befellegzett...”, „Az ígért színes gépek gyártása el sem kezdődött rendelni sem lehet...”, „A Primo céget föl lehet számolni...”. Hónapok óta különböző pletykák terjednek a sokak által szidott, sokak által kedvelt hazai mikrogépről. Gondoltuk olvasóink kedvére teszünk, ha megtudjuk mi az igazság.

Mielőtt elkezdenénk köntörfalazni, hogy igaz is meg nem is, meg így van meg úgy van, szögezzük le: a tények azt mutatják, hogy az eddigi legnagyobb hazai példányszámú mikroszámítógép gyártása leállt, s vajmi kevés az esély rá, hogy valaha még újra induljon. A „csöd”, tehát valóban „csöd”. Az igazsághoz az is hozzátartozik persze, hogy a Primo céget nem kell felszámolni, minthogy ilyen cég nincs és nem is volt soha. A Primo gép gyártására több cég adta össze anyagi és szellemi erejét. A kezdetben MICROKEY-nek, majd COSY-nak átkeresztelt mérnöki iroda részvényesei az EMO (Elektromodul) az MTA SZTAKI és a Sársápi új Élet Mgtsz.

Móricz Sándor a COSY igazgatója, Lebek István az EMO kereskedelmi vezérigazgató-helyettese és a Primo márkamenedzsere Pogrányi Károly nagy örömeinkre első szóra vállalták, hogy részt vesznek egy beszélgetésen, amelynek ezt a címet adhattuk volna: „Miért jutott ide a Primo?”.

Talán üzleti, taktikai, talán presztízs megfontolásból, a COSY vezetői a tavalyi év végén még legfeljebb csak vállalati körben vallották be, hogy a Primo további gyártása azon áll vagy bukik, hogy hogyan szerepelnek a második iskolaszámítógép-pályázaton. Ennek ellenére szakmai berkekben ezt mindenki sejtette, s legalább annyian voltak a pályázat eredményhirdetését eilendrukkerként várók, mint a „szimpatizánsok”. (Mi az utóbbiak közé tartoztunk.)

Nyilván nemcsak mi, hanem maguk a Primot gyártók, menedzselők, árusítók is azt gondolták, hogy a pályázat eredménye, a színes nyomógombos géppel az általános iskolai kategóriában elért 2. és a középiskolai kategóriában elért első hely kihúzza szekerüket a sárból. Őszintén szólva ma is azt hisszük, hogy joggal gondolták ezt, hiszen a pályázat eredményhirdetésén minimum 5000, maximum 10 000 gépre szóló rendelésről esett szó. Lebek István szomorúan mondja: – A gépekhez szükséges alkatrészeket nem kis munkával előre beszereztük, s ezek ma is a raktárban vannak. Sokáig már nem kerülgetjük, ha 1–2 hónapon belül nem érkeznek meg a megrendelések, akkor felhasználjuk más termékekhez. Hát így... Az ígért megrendelések jószerivel még csak nem is csordogálnak. Az eddig befutottak száma még az ezret sem éri el.

Mi történt?

Annak idején, a második iskolaszámítógép-pályázat kiírásakor minden érdekelt arra számított, hogy a dolog ugyanúgy működik majd, mint az előző pályázat idején. Vagyis, hogy a pályázatot gondozó Tudományszervezési és Informatikai Intézet dönti majd el, hogy a központi pénzeket kinek adja, miből mennyit rendel. Így hát a Primo második és első helye

megnyugtatónak látszott, egy nagyobb megrendezés valóban kinézett. Csakhogy időközben kiderült, hogy a számítástechnikai kormányprogram megvalósítását is utóljárta a decentralizálás, meg a gazdasági szabályozás szele. S hogy, hogy nem a pénzek már nem a központi kalapban vannak. Egyesek szerint a pénzeket a megyéknek osztották szét, mások szerint ezek a pénzek csak elméletben léteznek. Az igazság kiderítését egy későbbi lapszámunkban majd megpróbáljuk.

Most legyen elég annyi, hogy a tények a következő központi keret minimális, s ezt az összeget a TII nem megrendelésekre, hanem a megyei vásárlások támogatására fordítja (olcsóbban vásárolhatnak a megyék gépet, mert az ár egy részét teszi be a kalapba a TII). Másrészt tény az is, hogy a megyék vásárlási kedve enyhén szálva változó. Láthatólag nincs annyi pénzük, amennyire mondtuk a Primo számított. A kevesebb pénzből pedig inkább Commodore-t rendelnek, mintsem Primot. Igaz Pogrányi Károly esküszik rá, hogy ha nem ő lenne a Primo márkamenedzser, akkor is jobb gépnek tartaná a Primot a Commodore-nál, de ezt a véleményét úgy tűnik nem sokan osztják. Az, hogy a Primot kitették a piaci viharoknak, s ezt nem jól állta, mégiscsak azt mutatja, hogy az elmúlt másfél évben a Primot nem sikerült megszerettetni a vásárlókkal, a közvéleménnyel.

Lehet, hogy épp ez a „csőd” oka?

Labek István: – Az elején sok baj volt. Nagyon sok volt a hibás gép, ez rontott a renoménkon. Pedig utána egyenletes jó színvonalat produkáltunk.

Móricz Sándor: – Azt hittük, hogy kis szériában „manufaktúrális körülmények” közt is lehet világszínvonalú technikát gyártani. Ma már tudjuk, tévedtünk.

Pogrányi Károly: – Az az igazság, hogy a Primo jó gép. A legtöbben a billentyűzetre mondták, hogy hibás, nem működik. Pedig csak egyszerűen be kellett volna a kívánt érzékenységre állítani.

BIT-LET: – Véleményünk szerint a gépet nem szerették meg a számítógépesek. Ha egy iskolában válogatni lehet a gépek közt, nem a Primoért verkednek a diákok. Hogy ennek a billentyűzet az oka vagy más, teljesen mindegy. Egyébként is hiba volt 1984-ben ilyen billentyűzettel megjelenni a Spectrumhoz, Commodore-hoz, HT-hez szokott hazai piacon. Kérdezhetik az olvasók, hogy akkor mi miért irtunk olyan jókat néhány hónappal ezelőtti különleges Vállalónkban a nyomógombos, színes Pro Primo-ról? Azért, mert ez a Primo valóban jól vizsgázott. S az az érzésünk, hogy a megrendelések többek közt azért maradtak el, mert ezt a gépet nem ismerik, nem hiszik el róla, hogy valóban sokkal jobb a réginél. A mi szívünk őszintén szálva vérzik ezért a jobb sorsra érdemes konstrukcióért. Hogy ugyanolyan jók lettek volna-e a sorozatgyártásban készülő gépek mint az elkészült néhány mintapéldány – úgy tűnik már soha sem fog kiderülni.

És most mi lesz?

A Primo azért nem szűnik meg létezni. Annál is kevésbé, mert pillanathyilag kb. háromezer gép van még az EMO raktáraiban. Ezek egy része már nyomógombos, ha nem is színes. Ezeket mindenképpen el szeretnék adni. Szeptember elsejétől közel 30%-kal olcsóbban lehet megkapni ezeket a gépeket. Az árak a következők: A32 – 5660 Ft; A48 – 7670 Ft; A64 – 8570 Ft; B32 – 7180 Ft; B64 – 12 590 Ft (A B jelű gépek nyomógombosak.) Ráadásul az iskolák a szűkös központi pénzalap támogatásával (Kettőt fizet – hármát kap módon) vásárolhatnak.

Apropós, ár. A Primo – ezt mindenki tudja – nagyon sikeresen indult még a meglehetősen sok műszaki hiba ellenére is. Hiszen annak idején 1984 őszén a legolcsóbb itthon kapható mikrogép volt. Azután

a helyzet pillanatok alatt megváltozott. Beszélgetőpartnereink elmondták, őket bizony felkészületlenül érte, hogy a kinti arverseny oly hamar elért hozzánk is. Nem tudtak mit csinálni. Az induló árhoz túlságosan sokáig ragaszkodtak, hiszen már a kezdet kezdetén nagyon kis nyereséggel kalkuláltak. Ma már tudják, hogy jövőjük érdekében mégis hamarabb kellett volna árat csökkenteni. A Primo sztori egyik tanulságaként ismét **hebizonyosodott, hogy a hazai ipar lehetőségei, az adózási, szabályozói rendszer nem teszik lehetővé, hogy egy hazai gyártmány árban versenyképes legyen egy nagy világcég termékével.** (Különösen ha ráadásul „dömpingáron” adott termékről van szó!) Nem csoda tehát, ha a Primo ügy megosztotta az ipari, oktatásügyi vezetőket. Az egyik fél váltig hangoztatta a hazai ipar támogatásának fontosságát, míg az ellentábor a piaci törvényszerűségek érvényesülését erőltette. Elismerve az utóbbi fontosságát, azért meg kell jegyeznünk, hogy valóban furcsa helyzet amikor egy országban a külföldi ipari termékek importja nagyobb kedvezményeket élvez, mint a hazai gyártó. Márpedig a Commodore-Pro-Primo „versenyben” ilyen helyzet állt elő.

Mi a magunk részéről ezúton veszünk búcsút egy jobb sorsra érdemes magyar mikrogéptől. Sajnáljuk, hogy ilyen rövid életű volt. Csodákban már nem reménykedünk, ha a színes gépre eddig 800 megrendelést kapott a cég, szerintünk már nem sokkal kap többet a következő 1–2 hónapban. A magunk részéről úgy gondoljuk, hogy rengeteg tanulságot rejt magában a Primo sztori, hogy pontosan milyeneket – erről a hazai ipar vezetőinek is érdemes lenne konzultálni az illetékesekkel. (Azaz beszélgetőpartnereinkkel.) És végül a magunk részéről persze nem tekintjük befejezettnek a Primoval való foglalkozást, mert egy mikrogép amelyből több ezer van az országban, nekünk még sokáig téma marad. Tehát Primo tulajdonosok figyelem! A BIT-LET-ben továbbra is számíthatnak programokra, információkra!



KERAVILL MEV

µELEKTRONIKAI

MÁRKABOLT 

BP. V. MÚZEUM KRT. 11.

MIKROELEKTRONIKA:

A JÖVŐ A JELENBEN.

★★★★★★★★★★★★★★★★★★★★

FÉLVEZETŐK,

INTEGRÁLT ÁRAMKÖRÖK,

MIKROPROCESSZOROK

ÉS CSATLAKOZÓIK.

SZAKTANÁCSADÁS, CSOMAGKÜLDŐ SZOLGÁLTATÁS

PROGRAM AJÁNLAT

Cirill betűk
a PRIMO-n

```

10 REM *****
20 REM *   O R O S Z   B E T Ű K   *
30 REM *****
40 POKE 16561,71,103: CLEAR 50: DEFINT
  I-P: DEFSTR U-Z: P=16452: DIM K(58):
  POKE P,128: POKE P+7,72,103
50 FOR I=0 TO 58: READ K: K(I)=K:
  BEEP K,30: NEXT: PRINT CHR$(2)
60 POKE 26440,0,112,16,16,24,20,20,25,0,
  68,68,68,100,84,84,101,0,60,68,68,60,
  20,36,69,0,68,68,68,60,4,4,5,0,84,84,
  56,16,56,84,85,0,112,8,4,60,4,8,113,
  40,124,64,64,112,64,64,125,0,124,64,
  64,120,68,68,121
70 POKE 26504,0,72,72,72,72,72,72,124,5,
  0,0,0,0,0,0,0,0,28,20,20,36,36,124,
  67,0,0,0,0,0,0,0,16,56,84,84,84,
  84,56,17,0,120,68,64,64,64,64,65,0,
  68,68,76,84,100,68,69,16,84,68,76,
  84,100,68,69
80 POKE 26568,0,28,20,36,36,36,36,69,0,
  124,68,68,68,68,68,69,0,72,84,84,116,
  84,84,73,0,84,84,84,84,84,84,125,0,
  84,84,84,84,84,84,126,3,0,0,0,0,0,0,
  0,0,64,64,64,112,72,72,113
100 REM MINTAALKALMAZAS
110 FOR J=0 TO 11: READ U: CLS:
  GOSUB 1010
120 FOR I=0 TO 3000: NEXT: NEXT
130 END
1000 REM BETUKIIIRATAS
1010 K=LEN(U): PRINT$ 6,21-K,W:
1020 FOR L=1 TO K: V=MID$(U,L,1): M=ASC
  (V)-32: N=K(M): PRINT CHR$(N):
  BEEP N,30: NEXT: PRINT
1030 RETURN
9000 REM ADATOK (READ)
9010 DATA .32,33,34,128,129,130,131,39,
  40,41,132,43,44,45,46,47,48,49,50,
  51,52,53,54,55,56,57,58,59,60,61,
  133,63,134,65,135,136,138,69,140,
  141,88,142,143,75,144,77,72,79,145,
  146,80,147,84,89,66,148,67,150,151
9020 DATA ODIN,DVA,TRI,&ET$RE,P%TY,
  SEXTY,XEMY,VOXEMY,DEV%TY,DEC%TY,
  ODINNADCATY,DVENADCATY

```

1. lista

```

90 FOR J=26440 TO 26623: S=S+PEEK(J): NEXT
91 IF S=9670 THEN PRINT "HELYES 1"
92 FOR J=0 TO 58: T=T+K(J): NEXT
93 IF T=5104 THEN PRINT "HELYES 2"
94 STOP

```

2. lista

A PRIMO iskolai alkalmazását sokoldalúbbá tehetjük, ha orosz szöveget is írhatunk vele. Ebben nem a karakterkészítés jelenti a legnagyobb gondot (BIT-LET 15, 30), hanem az orosz szöveg kényelmes bevitele. Itt közölt programom a szövegbevitelnek egyszerű (lényegében fonetikus) módját alkalmazza. Egy egyszerű fogással sikerült azt is megoldani, hogy a gép vegyesen használja az újonnan generált jeleket a latin ábécé felhasználható betűivel.

A szöveg bevitele (INPUT vagy DATA) közben ki kellett zárni a kettős betűket, ugyanakkor bevonni a munkába néhány különleges jelet (3. lista).

A mellékelt program 40-es sora a szükséges tárolóátrendezéseket végzi (védett terület az új karaktereknek, típusdefiníciók, tömbdimenzionálások, átállás nagybetűre). Az 50-es sorban történik egy nagyon fontos művelet: egy tömbben átrendezzük a karakterek eredeti kódjait, hogy a kívánt helyeken a gép az újonnan tervezett jeleket tudja írni. A 60-70-80-as sorokban töltjük helyükre az új karakterek adatait. A 100-as sortól tervezhetjük a tényleges munkát – most itt csak egy egyszerű demonstrációs programrész található, amely a képernyőre írja az orosz számneveket 1-12-ig.

Az 1000-es sortól található az orosz szó fonetikus alakját elemző és a ciril betűket kiírató szubrutin. A 9010-es sorban helyezkednek el az átrendezett kódok (fontos, hogy beolvasásuk még az előkészítő fázisban megtörténjék). A 9020-as sortól írhatjuk be az orosz szavakat. Ha a program szervezése olyan, hogy ismételt szükség lesz a beolvasásukra, ne felejtjük el alkalmas helyre beiktatni a RESTORE 9020 utasítást!

A program (1. lista) segítségével tetszőleges szóanyagú oktató, gyakorló vagy ellenőrző feladatokat állíthatunk össze.

A program beírása után érdemes ellenőrizni az adathalmaz helyességét. Iktassuk be a programba ideiglenesen a 2. lista sorait! Ha a próbafuttatás után megkapjuk mindkét „helyes” kiírást, törölhetjük ezt az öt sort (DELETE 90-94).

A program egyszeri futtatása után az új betűk a billentyűzetről is elérhetőek: CTRL+I + betű (A-V) billentyűzése orosz betűt ír a képernyőre. Programból a CHR\$(128) - CHR\$(151) utasításokkal lehet megjeleníteni az új karaktereket.

Fekete György 7300 Komló, Bocskai u. 35.

| | |
|-------|--------|
| # = Ъ | \$ = Ъ |
| % = Я | & = Ч |
| * = Ж | > = Э |
| É = Ё | Q = Ю |
| w = Ш | X = Ц |

Y = б

3. lista

Hasznos apróságok

Commodore 16

Olvastam Morvai László cikkét a BIT-LET júliusi számában, amelyben a C16 kiselbontású grafikájával foglalkozik. Remek cikk! A két karakterkészlet egyidejű megjelenítése például „életbevágóan” fontos volt a most készülő programomhoz, akár a többszínű karakterek lehetősége (bár ezt sejtettem). Én is felfedeztem néhány apróságot a C16-on, azt hiszem fontos segítség a szakirodalomban szegényes világunkban a programozóknak.

1. Bővített háttérszín mód

— Ez a lehetőség a C64-en is megtalálható. Ebben az üzemmódban a karakterek háttérszíne nem egységes, a COLOR1,... utasítással beállított szín. A képernyőmemória határozza meg a karakterek háttérszínét a következő elven: A 7–6. bit az aktuális pozíció háttérszíne, 5–0. bit az adott pozíción tárolt karakter kódja. Látjuk, hogy a karakterkód tárolására csak 6 bit áll rendelkezésre, ezért csak az első 64 karaktert használhatjuk ebben az üzemmódban. Tehát a SHIFT,C=RVS ON és ezek kombinációja a megfelelő háttérszínű karaktereket adnak. Eltűnik tehát a grafikus, és az inverz készlet (a kurzor is!). A 7–6. bit Morvai László által ismertetett módon határozza meg a színt (00 = háttérszín 01 = multi, 10 = multi 2, 11 = multi 3). Az egyes színeknek megfelelő címek FF15h, FF16h, FF17h, FF18h. Ezt az üzemmódot a FFO6 6. bitjének magasra állításával kapcsolható be: POKE65286, PEEK(65286)OR64; ki-
kapcsolása POKE65286, PEEK(65286) AND191.

2. Funkcióbillentyűk

— A funkció billentyűkhöz tartozó szövegeket a C16 a 055Fh–05E6h tárolja a következő elven: 055Fh címen tárolja az F1 billentyűkhöz rendelt szöveg hosszát, a 0560h címen az F2-höz rendelt szöveg hosszát tárolja,...,0566h címen az F8 billentyűkhöz rendelt szöveg hosszát tárolja. A valódi tároló 0567h-án kezdődik. Itt a hozzárendeléseket karakterenként a funkció billentyűk sorrendjében tárolja (1,2,3,...,8). Ha a nyolc mutatót, és a tárat kimásoljuk, akkor ezzel az egyszer beírt rendelkezéseket mentettük ki. Ajánlom a következő hozzárendelést a programfejlesztőknek:

```
KEY1,"RUN"+CHR$(13)
KEY2,"LIST"+CHR$(13)
KEY3,"AUTO10"+CHR$(13)
KEY4,"5 db cursor jobbra"
KEY5,"LIST1000"—+CHR$(13)
KEY6,"KEY5,"+CHR$(34)+"LIST 4 db space — 1 db
space"+CHR$(34)+"CTRL+2 :POKE 1413,13 CTRL+1
19 db cursor balra"
KEY7,"KEY7,"
KEY8,"HELP"+CHR$(13)
```

Ha lenyomjuk az F6-ot, akkor megjelenik a következő felirat: LIST —. A kurzor közvetlenül a LIST után áll. Ekkor definiálhatjuk, hogy melyik sortól listázzon az F5. Vizsgálják meg bátran, hogy mire szolgál a POKE utasítás a definícióban!

Fazekas Attila Kemece, Kecskéshegy 15. Pf. 56. 4501

A 64' er-ből!

Még az igazán jónak tartott C–16 BASIC is kiegészíthető hasznos POKE utasításokkal. Ehhez nyújt ötleteket a 64'er júliusi száma.

Az alábbiakban leírom a hivatkozott cikk „nem szó szerinti” fordítását. Az anyag közreadása bizonyára örömet okoz a C–16-osok tulajdonosainak.

POKE 205,0–24 A kurzort lefele mozgatja a megadott érték szerint.

POKE 775,128 Listavédelem. Csak a sorszám kerül kiadásra. (Normál értéke 139)

POKE 775,252 LIST utasítás hatására resetelődik a rendszer. (Normál értéke 139)

POKE 19,0 Engedélyezi az INPUT-kérdőjel kiírását.

POKE 19,1 Tiltja az INPUT-kérdőjel kiírását.

POKE 194,1 RVS ON

POKE 194,0 RVS OFF

POKE 239,0 Törli a billentyűzet puffert. Jól használható pl. a GET és GETKEY utasításoknál.

POKE 1525,0:1526,0:1527,0 SAVE védelmet eredményez. (A C–16 szimulálja a SAVE és a VERIFY utasításokat és minden úgy néz ki, mintha minden rendben lenne. Normálértéke: 255)

POKE 65286,PEEK(65286)AND239 Kikapcsolja a képernyőt. A programok ilyenkor kb. 6%-kal gyorsabban futnak. Praktikus a rendező programoknál vagy adathalmazból való keresésnél.

POKE 65286,PEEK(65286)OR16 Visszkapcsolja a képernyőt.

PRINT PEEK(56) * 256 + PEEK(55) – (PEEK(44) * 256 + PEEK(43))

Megadja a még rendelkezésünkre álló szabad memóriaterületet. (Nem a PRINT FRE(0) lekérdezést szimulálja.)

PRINT PEEK(56) * 256 + PEEK(55)

Megadja a BASIC tároló vétcímét +1-et.

PRINT PEEK(44) * 256 + PEEK(43)

Megadja a BASIC tároló kezdőcímét.

SYS 65526 vagy

SYS 65529 Resetet okoz.

SYS 32768 vagy

SYS 128 * 256 Minden esetben resetet okoz. A BASIC programot kitörli, a színek megváltoznak, a funkcióbillentyűk megtartják eredeti tartalmukat.

SYS 65499 A TI\$-et lenullázza.

És még egy tipp. Így lehet egy programot automatikusan betölteni és elindítani:

```
A$="LOAD"+CHR$(13)+"RUN"+CHR$(13)
FOR A=1 TO LEN(A$):POKE 1318+A,ASC(MID$(A$,
A,1)):NEXT:POKE 239,A:END
```

Honti Tamás 8000 Székesfehérvár, Kandó Kálmán Számítástechnikai Főiskola Kollégiuma Schönherz Z. u. 8.

Bármely program bonyolultsága

addig fokozódik,

amíg túl nem nő

programozója képességein!

(Murphy törvénykönyve)



K Ö N Y V M O L Y

Farkas Zoltán-Bálint András: **Commodore 64 filekezelés és input-output** – LSI ATSZ, 141 o., 119 Ft.

Azoknak nyújt segítséget a kötet, akik – bár a C 64 nem igazán alkalmas adatfeldolgozásra – jobb eszköz híján mégis ilyen feladatok ellátására kívánják használni gépüket. Az első rész a BASIC és a KERNAL I/O utasításait, gépi kódú rutinjait tartalmazza referenciaszinten, ismertetve a BASIC-bővítmések lehetőségeit is. A második rész a file-szervezéssel foglalkozik: egy adatnyilvántartó program részletes elemzésével mutatja be azokat a programozási módszereket, eljárásokat, melyek még nem haladják meg a gép kapacitását.

Dr. Jánoki Lajos–Dr. Kocsis János: **Számítógépes termelésirányítás** – Műszaki Könyvkiadó, 330 o., 90 Ft.

A szerzők az iparvállalatok termelés- és gazdaságirányítási részfeladatait, illetve azok kapcsolatát elemzik, jól hasznosítható ismereteket nyújtva e feladatok számítógéppel segített ellátásához az irányításban érintett vezetőknek, mérnököknek, közgazdászoknak, matematikusoknak.

Szlávi Péter–Zsakó László: **Módszeres programozás** – Műszaki Könyvkiadó, 116 o., 50 Ft.

A könyv hatékonyan működő, gyors, áttekinthető és könnyen módosítható BASIC-programokat elkészítésében segíti az olvasót. Weltner: **További tippek és trükkök a Commodore 64-eshez** – DATA BECKER – Novotrade, 187 o., 239 Ft.

A „Tippek és trükkök...” c. kötet folytatása – elődjéhez hasonlóan – házi használatra szánt ötletes programozási fogásokat mutat be. E könyv elsősorban a szoftvervédelemmel, a grafikával és a megszakításokkal foglalkozik, valamint áttekinti az operációs rendszer rutinjait és a KERNAL címtáblázatát. Hatodik fejezete a Lifes fiú című játék elkészítésének folyamatát írja le, a program teljes listájával együtt.

Székel László szerk.: **Sinclair Spectrum játék és program** – LSI ATSZ, 248 o., 194 Ft.

A ZX Spectrum otthoni felhasználóinak – játék, és egyszerű programozási, adatnyilvántartási célra – szolgál számos hasznos információval a kötet.

Szlávi Péter–Zsakó László: **Módszeres programozás** – Műszaki Könyvkiadó, 116 o., 50 Ft.

Nyilván mindenki járt már úgy egy feladattal találkozva, hogy azt hitte, tökéletesen érti azt. Ekkor leült a gép elé, hogy azzal keresse a megoldást – és egyre inkább belebonyolódott a program részleteibe. Amikor pedig verejtékes munkával végül is elkészült a „nagy mű”, kiderült, hogy nem az eredeti feladatot oldja meg, vagy nem úgy, ahogy kellene, vagy

olyan lassan, hogy nem érdemes kívánni, vagy... De tegyük fel, hogy esetleg mégis jól működött. Ám amikor egy hét múlva hasonló probléma megoldására is előveszünk, kiderül, hogy lehetetlen hozzáigazítani az új feladathoz, mert szerkezete – ha van egyáltalán – annyira áttekinthetetlen, hogy egyszerűbb az egészet előlről kezdeni.

Mi lehet ennek az oka? Kézbe véve a legtöbb programozással foglalkozó könyvet, általában az adott nyelv utasításainak felsorolását találjuk benne, jobb esetben fontosság, illetve tanulhatóság szerinti sorrendben, rosszabb esetben lexikonszerűen, ABC-rendben. Még a legdidaktikusabb kézikönyvek is, miután a programnyelv néhány közvetlen üzemmodú parancsát ismertetik, rögtön rátérnek a szintaktika fokozatos bemutatására. A kezdő felhasználó pedig, amikor boldogan látja, hogy a gép engedelmessé válik neki, könnyen azt hiheti, hogy a nyelv ismeretében bármit meg tud oldani, és olyan rossz programozási szokásokat vesz fel, amelyek az előbbi, kusza programokhoz vezetnek.

A BASIC a legkevésbé sem strukturált programnyelv. Paradox helyzet: a kezdő programozó BASIC-et tanul a könyvekből, azután más nyelvekkel megismerkedve alig tud szabadulni a hibás beidegződésektől. Azok a könyvek pedig, amelyek kicsit is magasabb szintű, jobban szervezett – vagy használjuk a kifejezést: módszeres – programozást tanítanak, már feltételeznek bizonyos számítástechnikai ismereteket, nyelvezetük is a szakzsargon.

Azt, hogy BASIC-ben is lehet jól szervezett, áttekinthető és gyors programot írni, Szlávi Péter és Zsakó László most megjelent kötete alapján láthatja be az olvasó. A kötet címe túl komoly. Tartalma viszont nemcsak a beavatottakat érinti, hiszen főleg azoknak íródott, akik otthon használják mikroszámítógépüket. A szerzők célja: egyetlen alapeladattal megoldására írt programot fokozatosan fejlesztve megmutatni, hogy miért jobb a módszeres megírt program, mint az, amely ad hoc (a „frontális támadás módszerével”) készül. Időnként el-elkanyarodnak a kitűzött feladattól, hogy az olvasót az ahhoz nem szorosan kapcsolódó algoritmusokkal, fogásokkal is megismertessék: de ezt a kacsarinognak tűnő utat példás logikai felépítéssel követik végig.

Ami az első belelapozáskor a legszembeötlőbb: a kódolással – vagyis a programnak az adott nyelven való leírásával, a géppel való megértésével – csak a programkészítés legvégső fázisaként foglalkoznak a szerzők. Egyetlen programnyelv használata helyett egy kiterjesztett, általánosított nyelvet vezetnek be, így biztosítva a legnagyobb függetlenséget az adott géptípustól.

A kötet nincs fejezetekre tagolva, sorszám

nélküli alcímek jelzik csak a következő néhány oldal témáját. Mégis elkülöníthetőek a fő részek. A szerzők először bemutatják a módszerek nélküli programírás legfontosabb hibáit, majd azokat az elveket ismertetik, amelyeken a módszeres programkészítés alapul. Ezt követi a legfontosabb algoritmusok leírása, melyek még a legegyszerűbb programozási feladatoknál is szükségesek lehetnek. A programhelyesség és -hatékonyság, a hibakeresés és a programtesztelés a következő témakörök; majd a programdokumentációról szóló rövid ismertetés zárja a kötetet. A terjedelmes függelék tartalmazza a szerzők által bevezetett általános programnyelv leírását, e nyelv kódolási szabályait öt elterjedt gép – ABC 80, HT 1080Z, ZX 81, ZX Spectrum és C 64 – BASIC-nyelvére, valamint a kötet elején kitűzött feladat megoldására szolgáló program listáit e gépekre.

A szerzők stílusa többnyire olvasmányos, nem tudományoskodó – kivéve, sajnos, a legfontosabbat: az alcímeket és az új fogalmak megnevezéseit. Ez lehet a legfőbb kifogás a könyvvel szemben. Hiszen a gyanútlan olvasó, ha nem riad vissza a komolykodó címtől, és belelapoz a könyvbe – vagy akár csak a hátoldalon kiemelt alcímeket futja át – nem jön rá, hogy a kötet az otthoni, amatőr programozóknak szól, nem a számítástechnikai szakembereknek. Visszateszi a könyvet a polcra – pedig hasznos olvasmány lenne.

Székel László szerk.: **Sinclair Spectrum játék és program** – LSI ATSZ, 248 o., 194 Ft.

A Spectrum az egyetlen a Magyarországon elterjedt mikrogépek közül, amely teljes joggal játékgépnek nevezhető. Bár a C 64 is ilyen céllal került kifejlesztésre – játékgéppel ellátottsága jobb is –, de a legsikeresebb Commodore-gép egyéb, szakmai célú szoftverkinálata is hatalmas, így vállalatoknál, intézményeknél is széles körben használatos. A többi mikrogép pedig főleg az iskolai oktatásban kap szerepet.

A Spectrumra írt programok játékellegének – illetve a más célú szoftverek szinte teljes hiányának – köszönhető, hogy a gép főfelhasználási köre a 10–16 évesek korosztálya. Legtöbbjük ha egy-két évig használja gépét, eljut arra a szintre, hogy nem másol át kritika nélkül minden játékot. Azokat a szoftvereket kezdi értékelni, amelyekben van valamilyen új ötlet, amelyek mások, mint a hagyományos úrháborúk, amelyeknek szép a grafikája és a hangja. Programozgat BASIC-ben, és keresi az egyszerű felhasználói programokat – BASIC-bővítméseket, szövegszerkesztőket, rajzolóprogramokat.

Az a játék-, illetve elemi segédprogram viszont, amely nyújt valami újat, nem biztos, hogy rögtön a betöltés után megérthető, a billentyűzettel vagy a joystickkal kísérletezve.

E programok kalózmásolatok útján terjednek, leírás nélkül. Emiatt (volna) alapmű az LSI ATSZ új kiadványa, mely számos Spectrum-játékot és -segédprogramot ír le.

Volna, mert annak ellenére, hogy – egy rövid, bevezető fejezet után – a második rész lexikonja ezeregy játékot sorol fel és osztályoz minősége szerint, a harmadik fejezetben sok olyan program kerül többé-kevésbé részletes bemutatásra, melynek osztályzata hármask vagy négyes; és kimarad számos, ötös minősítést elért, igazán jó játék.

Az osztályozás, persze, egy-egy program tulajdonságainak átlagát tükrözi, tartalmazhatja az egyszerűség, a könnyű kezelhetőség, a jó grafika együttes értékelését. A részletes ismertetésre való kiválasztás szempontjai viszont ezt figyelembe véve is megkérdőjelezhetők: helyet kapott itt sok, egymástól alig különböző, és semmiféle leírást nem igénylő játék is. Ráadásul az ismertetések színvonalra is meglehetősen eltérő. Némelyiken érződik, hogy írója sokat próbálgatta az adott programot, alaposan kiismerte, így taktikai tanácsokat is tud adni – mások pedig mintha a gyári leírás tömörített nyersfordításai lennének.

Jó, hogy a játéklexikon és -leírások a kötet fél terjedelmét sem teszik ki. Ami igazán értékes e könyvben, az ezután következik.

Még a játékoknál maradva: nagyon hasznos a HURG c. játékkészítő program referencia-szintű leírása a harmadik fejezet folytatásában. A kezdő felhasználó is megérti, hogyan kell akciójátékot írni – de nem ez a legfontosabb. Megtanul valamit abból is, hogy hogyan épül fel egy több részfunkciót teljesítő program, hogy mit jelent a menütechnika. És ha ezt átlátja, képes lesz annak felismerésére is, hogy melyik program az, amelyik tényleg tartalmaz valamilyen újdonságot, és melyik a régebbiek sablonos variációja.

A negyedik fejezet bevezetéseként két, jól használható rutin listája: kétszatosnás hanggenerátor és spritekezelő. A hanggenerátor tényleg érdekes hatásokra képes – rengeteg kísérletezés és gyakorlás után. A spritekezelő leírása egyértelmű – de hiányzik a gépi kódú rész assemblerlistája, nehéz a hosszú DATA-sorok hibátlan begépelése.

Ezután elterjedt, de leírás hiányában ki nem használt segédprogramok ismertetései következnek – grafika, BASIC-bővítések, a gépi kódú programozást könnyítő programok. Némelyik bő lére eresztett, mások szűkszavúak, de tényleg ezek a legelterjedtebbek. A BETA BASIC 3.0 (legterjedelmesebb bemutatás) pedig Spectrum-viszonylatban valóban csodákra képes. A „The Artist” szintén – csak hogy itt éppen a program lényegét jelentő fóliafunkció szerepe nincs kellően hangsúlyozva.

Az utolsó fejezet egy adatnyilvántartó program készítését mutatja be. Példaként a Forma I. futamok adatait kezeli, de persze használható másra is. A leírás fő erénye, hogy tudatosítja a kezdő felhasználót: a programírást nem lehet előzetes tervezés nélkül elkezdni. Maga a program nem képes túl sokra, hiszen a BASIC és a Spectrum sem igazán alkalmas ilyen feladatok ellátására.

A fenntartások ellenére is hiánypótló a kötet. Sok felhasználó számára ez lehet az egyetlen forrás, ahonnan a programjainak kezeléséhez szükséges információkat megszerezheti. A szerkesztő és a szerzők sokat változtattak a könyv elődjéhez (1001 játék és Graphics BASIC C 64-re) képest, ami e kötet előnyére vált. Kiegészültyezett szerkezettel, kevésbé ötletszerű programválogatással a sorozat következő darabja tényleg tökéletesen használható lesz – de hol az a harmadik, ennyire elterjedt géptípus, amelyről az a könyv szólhat?

Tallér József

POSTA

Olvasóink észrevételei alapján közöljük lapunk 1985. november 28. (179.) számában megjelent PRIMO billentyűtáblázat kiegészítését.

Az eredeti táblázatban szerepeltek a CTR+, és a Shift+, vezérlő billentyűkkel egy gombnyomással megadható kulcsszavak. Most megadjuk a + az ABC valamelyik betűjének együttes megnyomásával elérhető kulcsszavakat.

| | | |
|----------|----------|-------------|
| a - COS | g - CVS | n - MKDS |
| b - SIN | h - CVD | o - CINT |
| c - TAN | i - EOF | p - CSNG |
| d - ATN | j - LOC | q - CDBL |
| e - PEEK | k - LOF | r - FIX |
| f - EXP | l - MKIS | s - LEN |
| g - CVI | m - MKSS | t - STRS |
| | | u - VAL |
| | | v - ASC |
| | | w - CHR\$ |
| | | x - LEFT\$ |
| | | y - RIGHT\$ |
| | | z - MID\$ |

Amint az a táblázatból látható, itt már a továbbfejlesztést szolgáló utasítások kulcsszavai is megtalálhatók.

Néhány olvasónk felhívta a figyelmünket, hogy négy billentyűhöz tartozó kulcszót nem adtunk meg. Ezek a következők:

| | | | |
|---------------|------|--------------|-----|
| CTR+, +BREAK | FOR | CTR+, +CLS | LET |
| CTR+, +RETURN | GOTO | CTR+, +SPACE | OUT |

Ezek a billentyűk a SHIFT+ vezérlőkarakterek hatására is ugyanezt a kódot adják.

Örömmel olvastam a BIT-LET májusi számában a TVC vallatását. Levelemben nem a vallatáshoz szeretnék hozzászólni, hanem a géppel kapcsolatban felmerült problémáimhoz segítséget kérni.

Két éve ismerkedtem meg a BASIC nyelvvel, és körülbelül egy éve a Z80 gépi kódjával. Előzőekben már hosszabb-rövidebb ideig dolgoztam ZX81, HT 1080Z, valamint ZX Spectrum gépeken. Mindhárom gépet egyaránt programoztam BASIC és assembly nyelven. A TVC gép BASIC nyelvű programozása nem esett nehezemre, de a gépi kódú tett megállapításuk már az első próbálkozásomnál beigazolódott. Ezzel kapcsolatban lennének kérdéseim:

1. Játékprogramoknál elengedhetetlen a botkormány alkalmazása. Milyen típusú botkormány használható, és melyik portról olvasható le a két külső botkormány?

2. Az alábbi programmal szerettem volna egy byte-ot megjeleníteni a képernyőn. Sikertelenül. Kérem javítsák azt ki, és magyarázzák meg miért nem működik.

```
10 GRAPHICS 16
```

```
15 POKE 33.0:POKE 34,32
```

```
20 FOR A=0 TO 14:READ B:POKE 8192+A,B:NEXT A
```

```
25 DATA 62,255,50,0,128,50,1,128,50,2,128,50,3,128,201
```

```
30 EXT 0
```

```
35 IF INKEY$="" THEN GOTO 35
```

```
40 LIST
```

A gépi kód mnemonikja: LD A,255

LD (8000),A

LD (8001),A

LD (8002),A

LD (8003),A

RET

3. Szeretnék egy egyszerű szintetizátorprogramot írni. Ehhez lenne szükségem a hang gépi kódú programozására, valamint az amplitúdó szabályozására.

4. Utolsó kérdésem. Csak gépi kódú programot, hogyan menthetek ki kazettára?

Előre is köszönöm segítségüket, és várom a lapban megjelenő TVC-vel kapcsolatos cikkeket, programokat.

Papp Tamás, 5600 Békéscsaba, Tanácsköztársaság út 24/28. I/18.

1. A TVC-hez Videoton gyártmányú, valamint minden Kempston-kompatibilis botkormány használható. A két botkormánynak nincs külön port-ja, úgy olvashatók le, mint a billentyűk: INKEY\$ és GET utasítással.

2. A gépi kódú program várt eredményét az alábbi BASIC sor adja:

```
10 GRAPHICS 4 FOR I=49152 TO 49155:POKE I,255:NEXT I
```

A továbbiakhoz a TVC-ről szóló augusztusi számunkban elkezdett cikksorozat ad segítséget.

3. A BASIC-ben használt SOUND utasítás gépi kódú megfelelője:

LD B,DURATION

LD C,VOLUME

LD DE,PITCH

RST 30h

DEFB 33h

Bővebbet a következő számokban megjelenő TVC-ről szóló cikkek fognak közölni.

4. A kazetta rutinok leírását is a TVC-ről szóló cikkek közlik.

ORIC 1-hez játék- és alkalmazói programokat keresek. Mivel a gép meglehetősen ritka Magyarországon, szoftvert beszerezni szinte lehetetlen. Mentségemre szolgáljon, hogy amikor 1984 elején Londonban megvettem, még nem lehetett tudni, hogy a cég tönkremegy és nem sikerül a Sinclair Spectrum nyomába erednie. (Igy hát a Sinclair szegődött az ORIC nyomába már ami a tönkrementést illeti – A szerk.) A BIT-LET és olvasói az utolsó remény...

Herváth Gábor 1063 Budapest, Bokányi Dezső u. 19. Tel.: 115-348



Gépnyerő 2. fordulójának

megoldásai (a 3. feladatból helyhiány miatt egyelőre csak kettő):

1. feladat megoldása: következőtéseinket, tehát mindent, amit tudunk az ABC kis betűvel fogjuk megjelölni, és később így hivatkozhatunk rájuk.

Az alapfeltevésekből:

a) $O = 0$, tehát más betű nem jelenthet 0-t.
b) $1 < C < R$, s emiatt +a) miatt $2 < C < 8$.

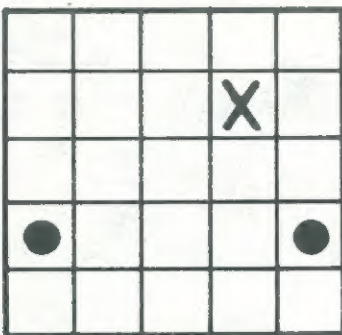
Az egyenlőséget átrendezve, az $A = A$ egyenlőséget és a) t felhasználva, valamint mindkét oldalról R-et és $100 \cdot I$ -t kivonva kapjuk, hogy

c) $PAP10 + RO + C16 = AP0SZ + SZE0$. Azonnal látszik, hogy d) $Z = 6$, s ezután mindkét oldalból 6-ot kivonva, majd 10-zel elosztva kapjuk az egyszerűbb:

e) $PAP1 + R + C1 = AP0S + S6E$ egyenlőséget.

Jelölje Σ az összeget, tehát e) bármelyik oldalát. Rögtön látható, hogy Σ első jegye vagy P, vagy P+1, de ekkor szükségképpen $A = 9$. Azonban ekkor $P = 8$ -nak kellene lennie,

1. ábra



s így a jobb oldal szerint $\Sigma > 9800$, de ezt a bal oldalon nem lehet elérni. Tehát Σ első jegye P, s ez a jobb oldalon ($P \neq A$) csak úgy teljesülhet, ha

f) $P = A + 1$. Ezután tehát P-t nem használunk, megjegyezzük, hogy g) ha A valamilyen értéket felvesz, akkor a többi betűnek nemcsak ezt, hanem az eggyel nagyobb értéket is el kell kerülni. S ebből és a)-ból d)-ből következik, hogy h) $1 < A < 8$ és $A = 5$, $A = 6$.

Most fejtjük ki az e) egyenlőséget, tudván, hogy C1-gyel a $10 \cdot C + 1$ számot jelöltük stb. Használjuk fel f)-et, majd rendezzük át:

i) $10 \cdot A + 10 \cdot C + 1 + R + 851 = 101 \cdot S + E$. Innen rögtön látható, hogy j) $S = 9$, ugyanis $S < 8$ esetén $101 \cdot S + E < 817 < 851$.

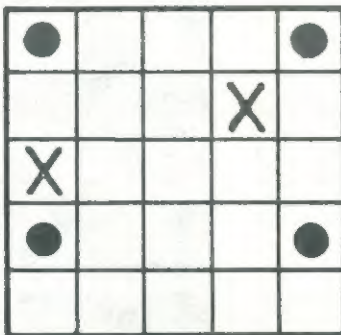
Ezután i) és j) alapján: k) $10 \cdot A + 10 \cdot C + 1 + R = 58 + E$.

a) és j) miatt $1 < E < 8$, ezért l) $59 < 58 + E < 66$.

Innen rögtön kapjuk, hogy m) $5 < A + C < 6$.

Először nézzük az $A + C = 5$ esetet! Ekkor k) és l) miatt n) $1 + R < 9$. $A = 5$ h) miatt, $A = 4$ b) miatt nem

2. ábra



lehet. Ha $A = 3$ lenne, akkor $C = 2$, s ezért b) miatt $1 = 1$, így n) miatt $R = 8$ és $E = 1$, azaz $E = 1$, ellentmondás. $A = 2$ esetén $C = 3$, s ez g)-nek mond ellent. Végül $A = 1$ esetén $C = 4$, s mivel a) és g) miatt 0, 1, 2 foglalt, ezért $1 = 3$, s így a legkisebb szabad érték 5, tehát $E < 5$; viszont ekkor $58 + E < 63$, ezért $R > 10$, ami lehetetlen, mivel R számjegy.

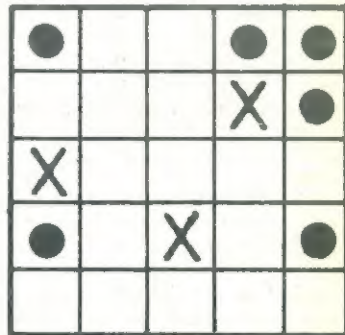
Tehát csakis o) $A + C = 6$ lehet. h) miatt $A = 5$. Ha $A = 4$, akkor $C = 2$, ezért $1 = 1$, l) miatt p) $1 + R < 6$.

Igy 0, 1, 2, 4, 5 foglaltsága miatt $R = 3$, de ekkor $E = 6$, azaz $E = Z$ lenne, ellentmondás. $A = 3$ esetén $C = 3$, azaz $C = A$ lenne. b) és p), valamint d) miatt $R < 5$ és $C < 4$, így $A = 1$ sem lehet. Maradt az utolsó lehetőség g) $A = 2$, s így r) $P = 3$. Ekkor s) $C = 4$, s így az előzőek miatt t) $R = 5$, és u) $1 = 1$. Ekkor v) $E = 8$, s így megkaptuk az egyetlen lehetséges megoldást: $32315 + 150 = 9685 + 416 = 23196$.

2. feladat megoldása:

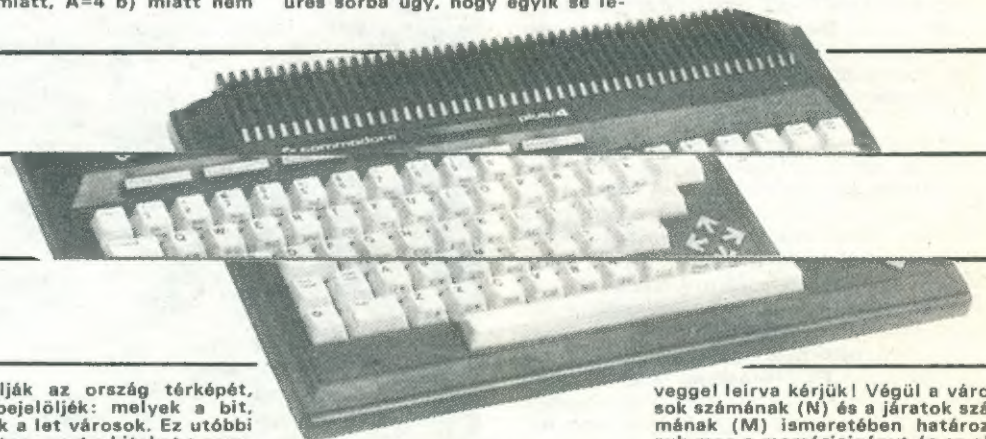
Legyen Második stratégiája a következő: Miután Kezdő lerakta első jelét, ő rakjon 2 jelet egy még üres sorba úgy, hogy egyik se le-

3. ábra



gyen a Kezdő jelével egy oszlopban! (1. ábra) Ezután Kezdő lerakja valahová a második jelét, majd Második rakjon megint két jelet egy üres sorba (eddig maximum 3 foglalt sor van) úgy, hogy jelei az eddig lerakott két jelével azonos oszlopba kerüljenek. (2. ábra). Ekkor 4 "vonal" van, melyben Másodiknak 2-2 jele található, s ezekből maximum 1 vonalban van már Kezdőnek is jele. (Igy csináltuk - egyszerű végiggondolni vagy -próbálgatni). Kezdő a következő lépésben maximum még egy "vonalat" tud elrontani, tehát legalább kettő szabadon marad. Így Második a 3. lépésével mindkét szabad vonalra rakhat 1-1 jelet (3. ábra), s így kezdő bármilyen válaszára (mivel ez a 4. lépés, ezért Kezdő nem nyerhet) ki tudja egészíteni valamelyik "vonalban" a jelei számát 5-re, s így mindenképpen nyer. Könnyű végiggondolni, hogy ha kezdő jól játszik, akkor Második a 3. lépésében még nem nyerhet, így ez a legjobb stratégia, vagy legalábbis egy a legjobb stratégiák közül.

harmad- gép nyerő



Harmadgépnyerő 3. feladat

Az Operációs tengeren van egy sziget, BIT-LET, ahol bitek és leték élnek. A bitek és leték nem keverednek, külön városokban élnek, de a bit és a let városok látszatra teljesen egyformák. A bit városok között is és a let városok között is nagy versengés folyik, ezért egy bit város lakója ki nem állhatja más bit városok lakóit, s hasonlóan egy let se bírja elviselni más városban lakó let társaságát, azonban bitek a letekkel jóban vannak (annak ellenére - vagy éppen azért? -, hogy nem keverednek egymással). A városok között vannak rendszeres repülőjáratok, de az eddigieknek megfelelően csakis bit városból let városba vagy let városból bit városba szállnak repülőik, két bit vagy két let város között elképzelhetetlen repülőjárat.

Commodoria meg akarja támadni BITLET-et, ezért repülő felderítőket küldött ki. Feladatuk, hogy

megrajzolják az ország térképét, s hogy bejelöljék: melyek a bit, és melyek a let városok. Ez utóbbi azért fontos, mert a biteket a commokkal, a letéket a dorokkal lehet elkábitani, azonban ha a biteket akarják dorokkal vagy a letéket commokkal legyőzni, akkor azok legyőzhetetlenné válnak.

A felderítők nem szállhatnak le, így csak a repülőjáratokat figyelhetik meg, ezenkívül egy besúgótól megtudták, hogy a fővárost, mely kétszer akkora, mint a többi város, bitek lakják. A felderítők egy, az alábbihoz hasonló vázlatot küldtek el Commodoria Főszámítógépének, hogy az megállapítsa: melyek a bit, és melyek a let városok. A rajzon a vonalak két város között a rendszeres repülőjáratokat jelentik. Segítsünk Commodoria Főprogramozójának! Tervezzük meg egy adatstruktúrát, mely segítségével lehet tárolni a felderítők adatait (mely az ábrához hasonló, de sokkal több várost és járatot is tartalmazhat!), s adjunk meg egy algoritmust, mely segítségével minél gyorsabban meg tudja majd a gép állapítani, hogy melyek a bit, és melyek a let városok.

Az algoritmus szűrje ki azt az esetet, amikor az adatok nyilván hibásak (tehát nem lehet a városokat 2 osztályba osztani úgy, hogy egy osztályon belüli két város között ne menjen repülőjárat), s azt az esetet is, amikor az adatok hiányosak, azaz nem lehet minden városról

el dönteni a milyenségét. Azonban ekkor is azokról a városokról, melyekről el tudja dönteni, hogy bit vagy let városok-e, ezt közölje. Commodoria Főszámítógépéről csak azt lehet tudni, hogy hasonló elven működik az általunk ismert gépekhez, a programozása előttünk ismeretlen, így az algoritmust szö-

veggel leírva kérjük! Végül a városok számának (N) és a járatok számának (M) ismeretében határozzuk meg a memóriaigényt és az algoritmusunk lépésszámát is! A jó algoritmus, mely nagy N-re és körülbelül $N \cdot N/8$ nagyságú M értékre a legkevesebb lépésben oldja meg a feladatot (átlagosan, ill. a legrosszabb esetben).

